

EVALUATION OF A HIGHLY SKEWED PROPELLER
FOR A NAVAL AUXILIARY (AO-177)

by

RICHARD L. JAMISON

LIEUTENANT, U.S. NAVY

S.B., Massachusetts Institute of Technology
(1976)

SUBMITTED TO THE DEPARTMENT OF OCEAN
ENGINEERING IN PARTIAL FULFILLMENT
OF THE DEGREE OF OCEAN ENGINEER
AND THE THE DEGREE OF MASTER OF
SCIENCE IN NAVAL ARCHITECTURE
AND MARINE ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1982

© Massachusetts Institute of Technology 1982

EVALUATION OF A HIGHLY SKEWED PROPELLER
FOR A NAVAL AUXILIARY (AO-177)

by

Richard L. Jamison

Submitted to the Department of Ocean Engineering on May 7, 1982, in partial fulfillment of the requirements for the degree of Ocean Engineer and the degree of Master of Science in Naval Architecture and Marine Engineering.

ABSTRACT

The cavitation performance of a seven-bladed, highly skewed propeller for a Naval auxiliary (AO-177) is evaluated using lifting surface numerical hydrodynamic methods. An important contribution to accurate prediction of cavitation performance is shown to be an accurate model of the effective wake.

A new model of the effective wake is described. A program to calculate unsteady time-averaged, but not circumferentially averaged, field point velocities is presented. The propeller-induced velocities, along with the original nominal wake, are combined with Huang's axisymmetric effective wake scheme in pie-shaped wake segments to determine the effective wake.

The method is used to predict severe cavitation extent. This prediction is confirmed by SSPA experiments.

Thesis Supervisor: David V. Burke

Title: Professor of Naval Architecture

ACKNOWLEDGEMENT

The author wishes to express his gratitude to the Navy for supporting this thesis. In particular, M. Wilson and R. Boswell of DTNSRDC and A. Chase of NAVSEA provided the needed raw data and background information in a timely manner and expressed strong support by way of their interest.

Thanks are extended to Professor D. V. Burke, who served as thesis advisor and who also provided the original motivation for the project, and to Professors J. E. Kerwin and R. Van Houten, whose experience and physical insight invariably got the project moving again each time it bogged down.

I would also like to express my appreciation to my wife, Beverly, not only for her support of our home, but also for her technical assistance in teaching me programming techniques and debugging tricks.

Finally, I wish to thank my daughter, Ruth, just for being Ruth.

TABLE OF CONTENTS

TITLE	PAGE
Abstract	2
Acknowledgement	3
Table of Contents	4
I. Introduction	5
II. Impact of Effective Wake	7
III. Method of Effective Wake Calculation	10
IV. AO-177 Effective Wake Calculations	15
V. Conclusions	19
VI. Recommendations	20
Appendices	
I. Nominal, Induced, and Effective Velocities	21
II. Nominal, Induced, and Effective Velocities, Non-Dimensionalized	28
III. Propeller Cavitation, Nominal Wake	35
IV. Propeller Cavitation, Effective Wake	44
V. Cavity Length, Nominal and Effective Wakes	53
VI. Cavity Volume, Nominal and Effective Wakes	55
VII. SSPA Experimental Propeller Cavitation Extent Plots	57
VIII. Nominal Wake Diagram	60
IX. Effective Wake Diagram	62
X. UFPV Program Listing	64
XI. PLEWAKE Program Listing	89
References	98

I. INTRODUCTION

The David W. Taylor Naval Ship Research and Development Center has recently designed a seven-bladed, highly skewed propeller for a Naval Auxiliary (AO-177). This design is instructive concerning the state of the art in computer-aided propeller design.

The history of the propeller design is chronicled in reports by Boswell (1), Valentine and Chase (2), and Hendrican and Remmers (3). Computer-aided propeller applications included lifting line theory, used to determine the radial load distribution and the radial hydrodynamic pitch angle, and lifting surface theory, used to determine the final geometric pitch distribution, camber distribution, and final propeller offsets, including fillets, trailing and leading edge details, additional thickness added to the trailing edge, tip geometry, rake, and hub details. Cavitation performance was predicted by the method of Burrill and Emerson (4) and was also evaluated by experiment.

However, sea trials of the AO-177 indicated a severe unsteady propeller force problem. Inspection of the propeller indicated that it was cavitating significantly worse than had been predicted. The Burrill and Emerson method, originally designed for

four-to-six-bladed propellers with an expanded area ratio of 0.60, did not extend well to the seven-bladed AO-177 propeller with an expanded area ratio of 0.77, even though both were for merchant-type hulls. Further, scale effects could have thrown off the model tests.

II. IMPACT OF EFFECTIVE WAKE

The AO-177 propeller performance was evaluated at MIT using some numerical propeller analysis computer programs developed at MIT. Kerwin and Lee (5) describe MIT-PUF-2, a computer program to calculate forces, steady and unsteady, generated by a propeller. MIT-FPV, a steady field point velocity program, is described in a report by Min (6). Lee (7) reports on MIT-PUF-3, a computer program to predict steady and unsteady propeller cavitation, extent and volume. All three programs model the propeller as a grid of discrete vortex segments. The wake is also modelled as a vortex grid in each program.

Cavitation performance calculations, more than propeller force and induced velocity calculations, are highly sensitive to the wake data used as input. Experience at MIT and elsewhere has shown that using the nominal wake data to approximate inflow conditions, while acceptable for propeller force and induced velocity calculations, is not acceptable for cavitation performance calculations. The wake velocities measured behind a model in the absence of any propeller (the nominal wake) must be modified to account for change in boundary conditions imposed by the presence of a propeller. This new, modified wake is generally called the "effective wake." There is, unfortunately, no

general method for anticipating the effective wake generated by a particular propeller in a general nominal wake.

Huang and others (8,9) have developed both theory and numerical schemes for calculating the effective wake of a propeller in an axisymmetric nominal wake. Det Norske Veritas has attempted to adapt Huang's method to non-axisymmetric wakes, such as surface ship wakes, by dividing the nominal wake into pie-shaped segments and then applying the axisymmetric wake calculation within each pie segment, using the nominal wake and the steady propeller induced velocities. This approach is an improvement over blindly applying the axisymmetric effective wake calculation to a surface ship's wake, but, owing to the very sharp AO-177 wake (even by surface ship standards), it was judged that some further refinement of Huang's method was required.

Huang's method assumed an axisymmetric nominal wake and a steady propeller-induced velocity field. The Det Norske Veritas method assumed a general nominal wake and a steady propeller-induced velocity field. Short of a completely general effective wake calculation, the most obvious refinement was to assume a general nominal wake, but allow an unsteady propeller-induced velocity field. This was the method

used for this paper.

The current method of effective wake calculations can be summarized in the following steps:

- 1) Use the nominal wake and propeller data to calculate the unsteady propeller-induced velocity field at representative points both inside and outside the propeller disk.

- 2) Divide the nominal wake into small pie-shaped segments as in the Det Norske Veritas method.

- 3) Calculate the effective wake velocities within each pie segment using the nominal wake velocities and the unsteady propeller-induced velocities, assuming that each pie segment acts as though it were part of an axisymmetric wake with the same radial velocity distribution.

III. METHOD OF EFFECTIVE WAKE CALCULATION

The two new aspects of the current effective wake calculations are the propeller-induced unsteady field point velocity (UFPV) calculation and the use of these time-averaged but not circumferentially-averaged velocities in a pie segment nominal wake modification calculation (PIEWAKE). Each of these two new methods will be illustrated by a typical calculation.

The unsteady field point velocity (UFPV) program is basically a generalization of the steady field point velocity program (MIT-FPV) by Min. (6). The unsteady version uses the full wake lattice arrangement also employed in MIT-PUF-2. Program inputs are the FILE14 DATA file output from MIT-PUF-2 and interactive directions. Program outputs are either single point calculations directed to the user's terminal or a data file to be used as input into PIEWAKE, for effective wake calculations.

UFPV first reads the FILE14 DATA input, containing mostly propeller geometry, singularity strengths, and singularity geometry. It should be noted that the number of propeller revolution time steps specified in MIT-PUF-2 must be divisible by the number of propeller blades. For effective wake calculations, the wake input to MIT-PUF-2 ought strictly to be the effective wake; however, using the nominal wake to calculate the propeller-induced velocities

is a good approximation.

The user next inputs field point coordinates. The field point position may be entered in cylindrical coordinates or in terms of the rotating propeller and wake grids.

The velocity induced by one blade at each time step is then calculated. At each time step, the vortex lattice is assigned the strengths corresponding to the time step of the propeller rotation. The velocity induced by each element of the vortex grid, both blade and wake, is then summed to obtain the total velocity induced by one blade in that particular angular position.

The velocities induced by each blade of the propeller are then summed to produce the total propeller-induced velocity for that position of the propeller. Velocities are calculated for different propeller positions and then harmonically analyzed. The zeroeth harmonic, then, is the time-averaged propeller-induced velocity for that field point. (By averaging all points at a given radius we could obtain the circumferentially-averaged, time-averaged, propeller-induced velocity, the output of the steady M1T-FPV.)

For effective wake calculations, such velocities are computed for up to 60 points per radius for radii varying from propeller hub to 1.7 times the propeller radius.

Between the propeller hub and propeller tip, the velocities are calculated at the leading edge panel on the propeller grid.

For the AO-177 propeller analysis, 56 points per radius for eight radii were used to create a file of unsteady field point velocities.

PIEWAKE is a program to perform Huang's axisymmetric effective wake contraction calculations in each of many pie-shaped segments of the wake. It uses nominal wake data and unsteady propeller-induced velocity data from UFPV as inputs to produce an effective wake velocity data file as output.

After reading nominal wake data and the induced velocity field data, PIEWAKE extrapolates the nominal wake data to the hub. The innermost radius of the induced velocity field is taken as the hub. (UFPV had taken, as its innermost radius, the hub radius used in MIT-PUF-2 and passed to UFPV by way of FILE14 data file.)

In each pie segment, Huang's method is used to calculate an effective velocity and the effective radius corresponding to that effective velocity. In the numerical form, this method is iterative in nature.

A finite difference equation, given in reference (9), is used to calculate the effective velocity. Given the

nominal wake at radii (assumed, for the first iteration, to be the nominal radii) in the presence of a propeller, an effective velocity can be calculated if an effective velocity at the next radius away from the hub is known. The effective velocity at the outermost radius is assumed to be equal to the nominal velocity there, enabling all effective velocities to be calculated.

A second finite difference equation, also given in reference (9), is used to calculate the effective radius corresponding to the set of effective velocities just calculated. This equation uses nominal radii, nominal velocities, and apparent velocities. (An apparent velocity is the sum of the effective velocity and the propeller-induced velocity at a point.) Given one effective radius, the next radius away from the hub can be calculated. Since the innermost radius of all velocity fields is set at the hub radius, all effective radii can be determined sequentially from innermost to outermost. These radii can then be used in the finite difference equation to calculate the next iteration of effective velocities. PIEWAKE iterates five times for each pie segment.

In addition to the effective velocity wake field, PIEWAKE also computes the mean velocity, the volumetric mean velocity of the nominal wake, the volumetric mean velocity of the effective wake, and the effective blockage.

The effective velocity field then be processed

and used by MIT-PUF-3 as any other wake field.

IV. AO-177 EFFECTIVE WAKE CALCULATIONS

Appendix I contains plots of nominal, induced, and effective velocities varying circumferentially at several specified radii. Appendix II contains the same information, except that the nominal and effective velocities are normalized on the respective volumetric mean velocity. These plots confirm that the modifications to the nominal wake outlined in Chapter III are indeed reasonable. Several salient conclusions can be drawn from these plots:

1) Effective velocity modifications of the nominal velocities are in the expected direction. Physically, the imposition of the propeller on the nominal wake draws more water into the slipstream, thus increasing the velocities. This "faster water" will come from radii greater than the field point being sampled; in other words, the wake will seem to "contract."

2) Effective velocity modifications of the nominal wake velocities are greatest at lower radii. The wake "contraction" will seem to concentrate here.

3) In general, the wake peak is more narrow in the effective wake than in the nominal wake. In the case of the AO-177 nominal wake, the existence in the presence of a propeller of a wake peak as sharp as that in the nominal wake is counter-intuitive. Therefore, a decrease in the wake peak is to be expected.

4) The normalized effective velocities at 0 degrees

are not appreciably different from that of the nominal wake. This is disappointing and was not expected. It may be a result of radial vorticity, ignored by the axisymmetric effective wake calculation performed in each pie segment. This assumption is least appropriate in the 0 degree region of the wake. A more general effective wake calculation scheme may be required to obtain better results in this region.

Results of MIT-PUF-3 cavitation performance of the AO-177 propeller are shown in Appendices III and IV. As an indicator of the influence of effective wake modifications to the nominal wake, MIT-PUF-3 was run twice, once with the nominal wake as input and once with the effective wake (the more accurate boundary condition) as input. Appendix III shows cavitation performance given the nominal wake as input while Appendix IV shows cavitation performance given the effective wake as input. In general, the plots show that the cavitation extent is not changed much, although cavitation performance is better in the effective wake at the lower radii where, as has been shown, the effective wake modifications are the greatest. At the outer radii, where the effective wake modification is the least significant, the cavitation performance is least changed between the nominal wake and effective wake cases.

Appendix V contains a graph comparing cavitation extent of MIT-PUF-3 output given nominal wake input against that given effective wake input. It plots the length of

cavitation on the seventh radial cavitation panel (about the .8 propeller radius) non-dimensionalized on the local chord length as the blade rotates through 360 degrees. This graph indicates that the effective wake modification did not reduce cavitation extent on a blade, a result noted earlier. However, a larger contrast between the nominal wake and effective wake cases is shown in Appendix VI which graphs the cavitation volume on a blade in the two cases. The volume is non-dimensionalized on the cube of the propeller radius. This graph shows a drop of approximately 35% of peak cavitation volume. This change in cavity volume demonstrates the sensitivity of cavitation performance to small changes in the wake profile. This confirms that efforts to refine effective wake calculations were indeed justified.

The effective wake cavitation performance indicated by MIT-PUF-3 compares reasonably well with experiments performed by the SSPA propeller tunnel. These experiments measured cavitation performance of the AO-177 propeller by photographing model tests. While no cavitation volume results were obtained, cavitation extent could be compared with MIT-PUF-3 predictions. Despite reductions of the effective wake calculations compared with nominal wake calculations, MIT-PUF-3 still tended to predict greater cavitation than the experiments indicated, particularly at the middle radii. This seems to indicate that the effective wake modifications, while tending to the proper direction, were not large enough. This is consistent with

the conclusions from comparing the two wakes that a more general effective wake scheme including radial vorticity in high wake regions is required.

V. CONCLUSIONS

1) Calculation of an effective wake using Huang's axisymmetric method in pie-shaped segment with unsteady propeller-induced velocities is a reasonable approximation. It still tends to underestimate the effective wake modification to the nominal wake in the vicinity of 0 degrees. To correct this, a more general effective wake calculation scheme including radial vorticity effects may be required.

2) The current effective wake calculation scheme can be used to predict cavitation performance for highly skewed propellers with up to seven blades.

3) Cavitation performance is indeed sensitive to small changes in the wake profile. Seemingly small improvements in effective wake calculations yield significantly more accurate cavitation performance predictions.

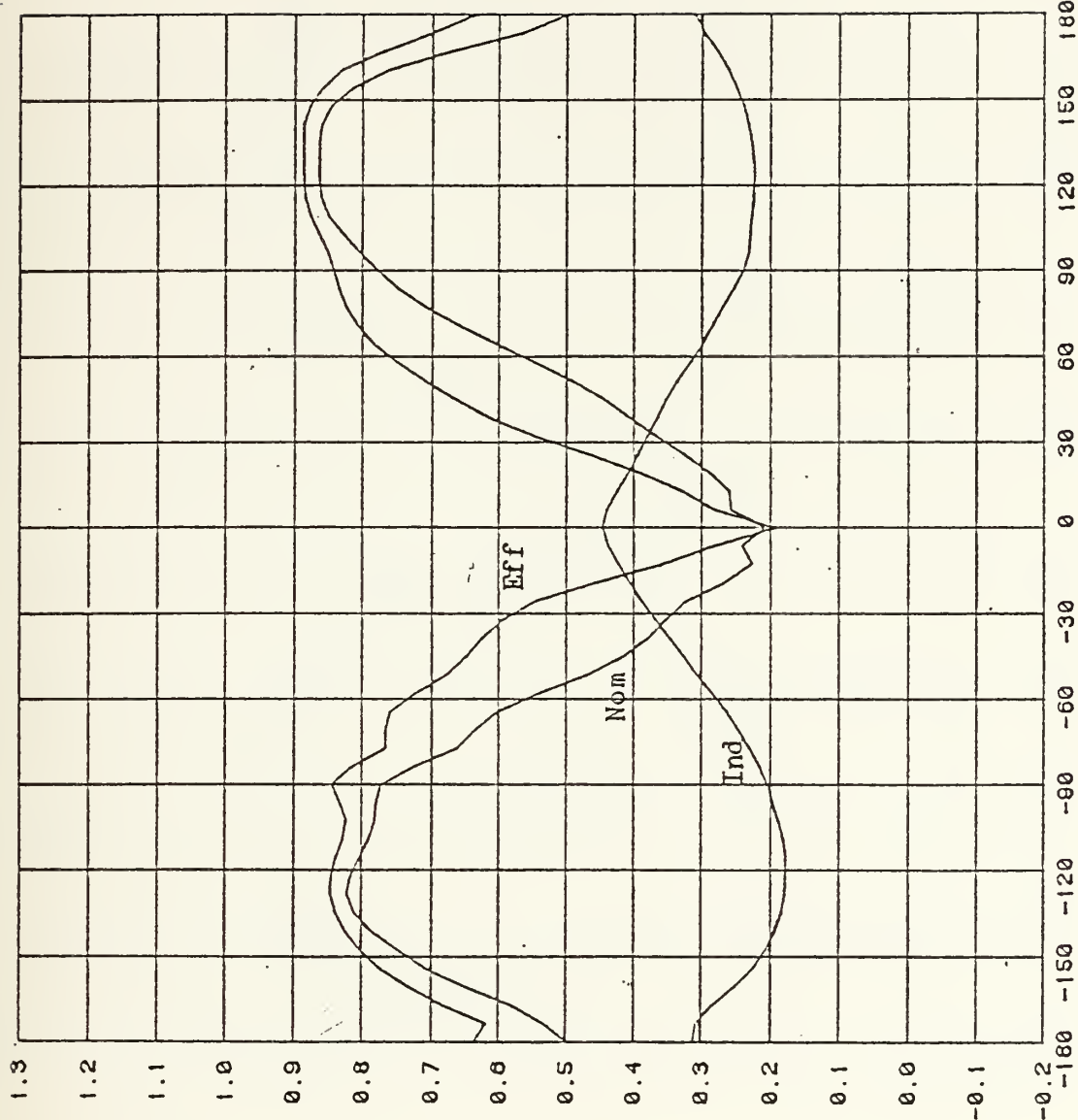
VI. RECOMMENDATIONS

1) A more general effective wake calculation scheme, including the effects of radial vorticity, is required.

2) The computational time (and hence cost) of the unsteady field point velocity program (UFPV) can probably be cut without significant sacrifice of accuracy by decreasing the number of points calculated per radius. Appendix I shows that the variation of induced velocity with rotation is slight and smooth. Calculation of fewer points and estimation of the remaining points using harmonic analysis would probably not reduce accuracy.

**** APPENDIX I ****

Nominal, Induced, and Effective Velocities



AXIAL WAKE V_{CRD}/V_{VOL}

$V(VOL)$

1.0

1.0

1.0

Nom

Ind

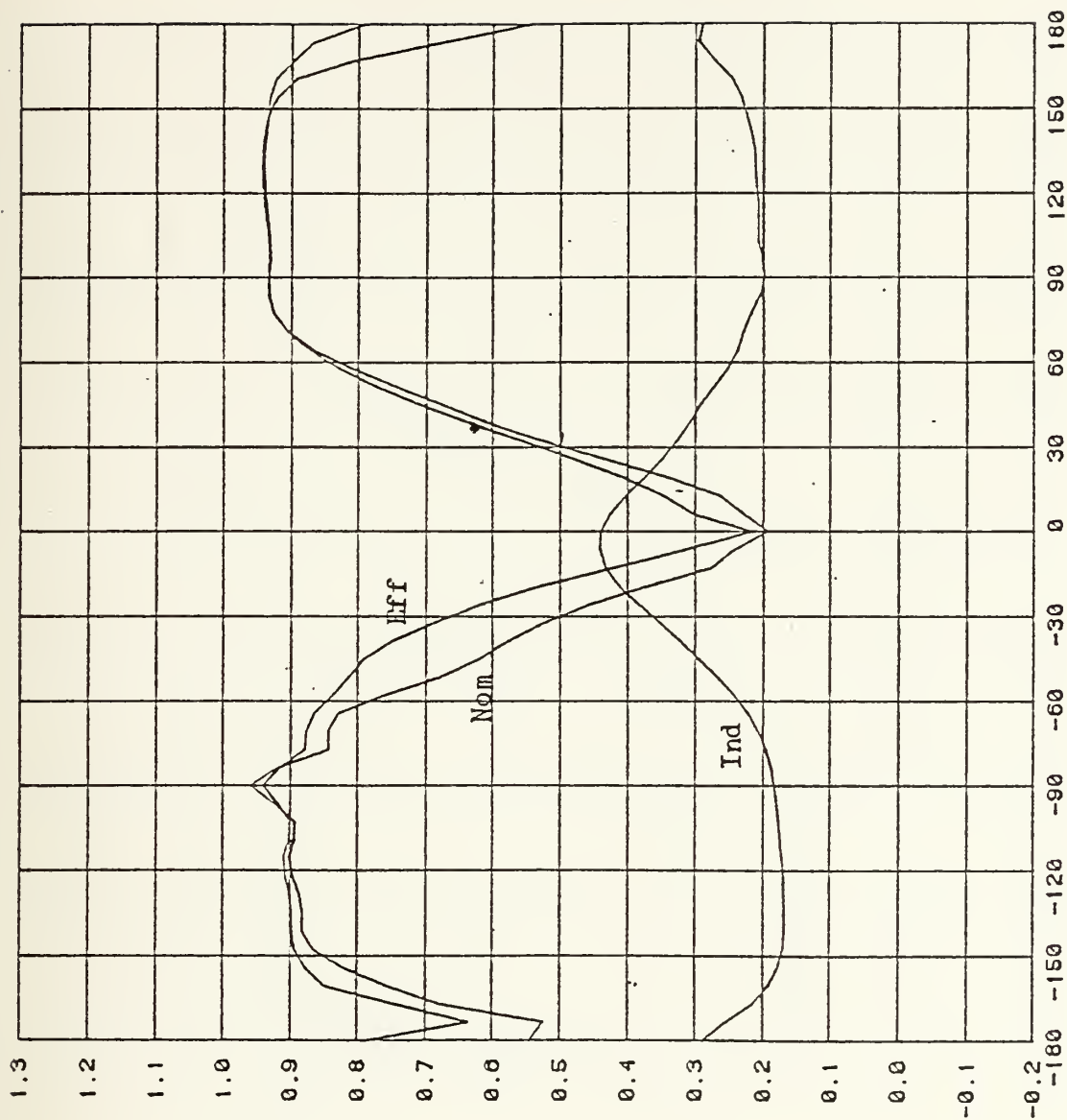
Eff

AXIAL WAKE VCR2/V(VOL)

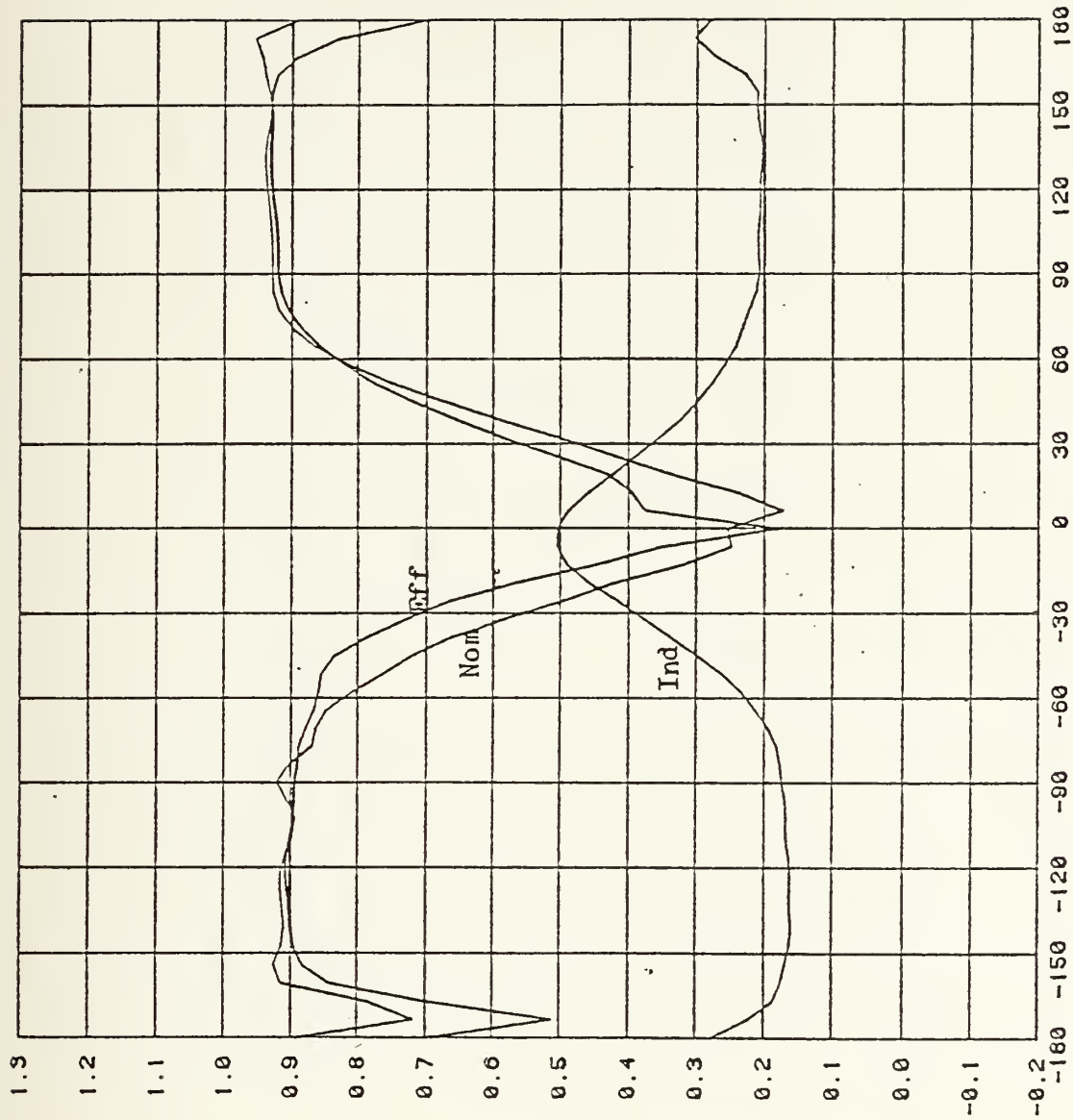
V(VOL)

Nom
Ind
Eff

1.0
1.0
1.0



NOM, IND, EFF VELOCITIES, .60 RADIUS



AXIAL WAKE V(CR)/V(VOL)

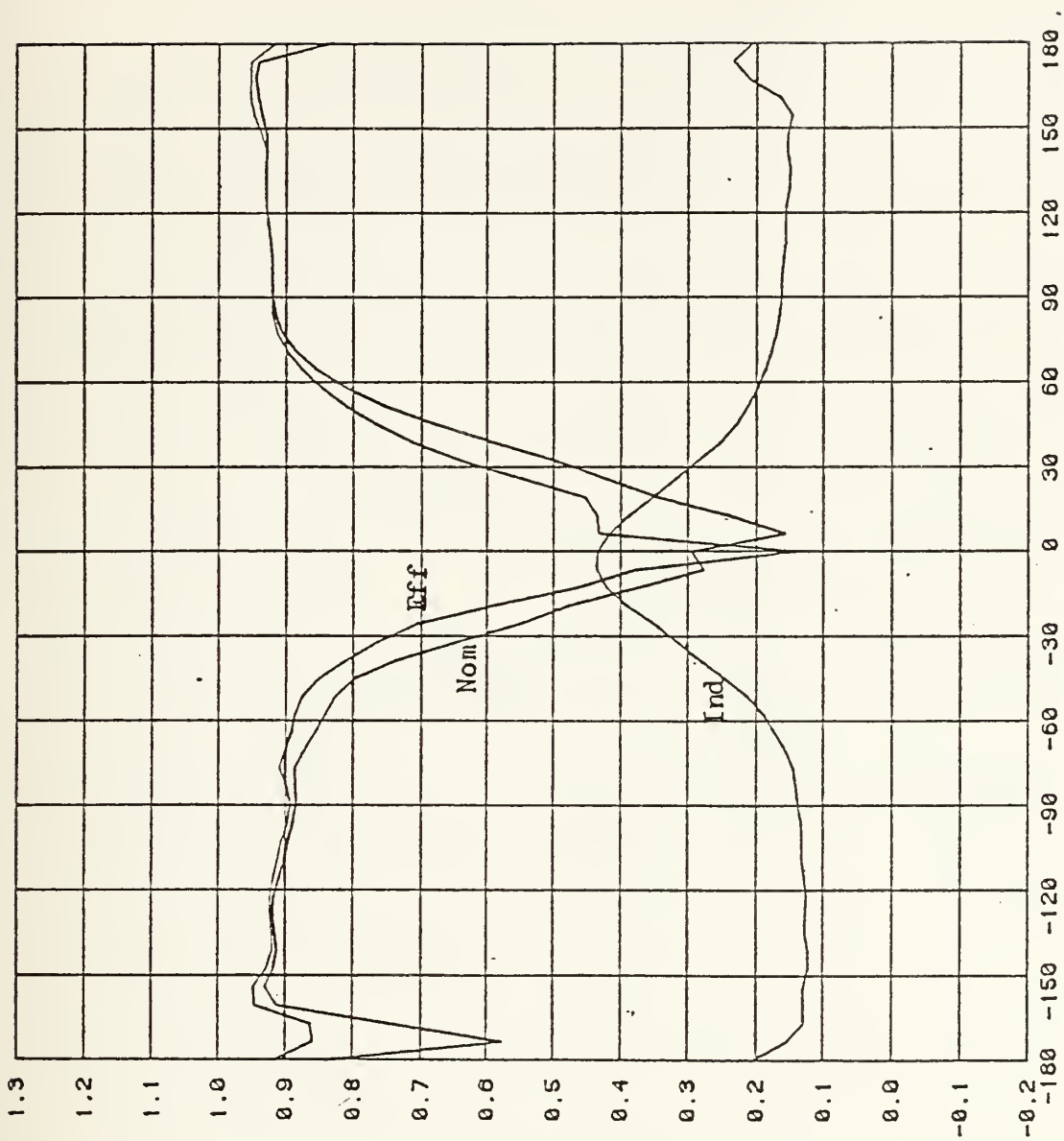
V(VOL)

Nom

Ind

Eff

NOM, IND, EFF VELOCITIES, .70 RADIUS

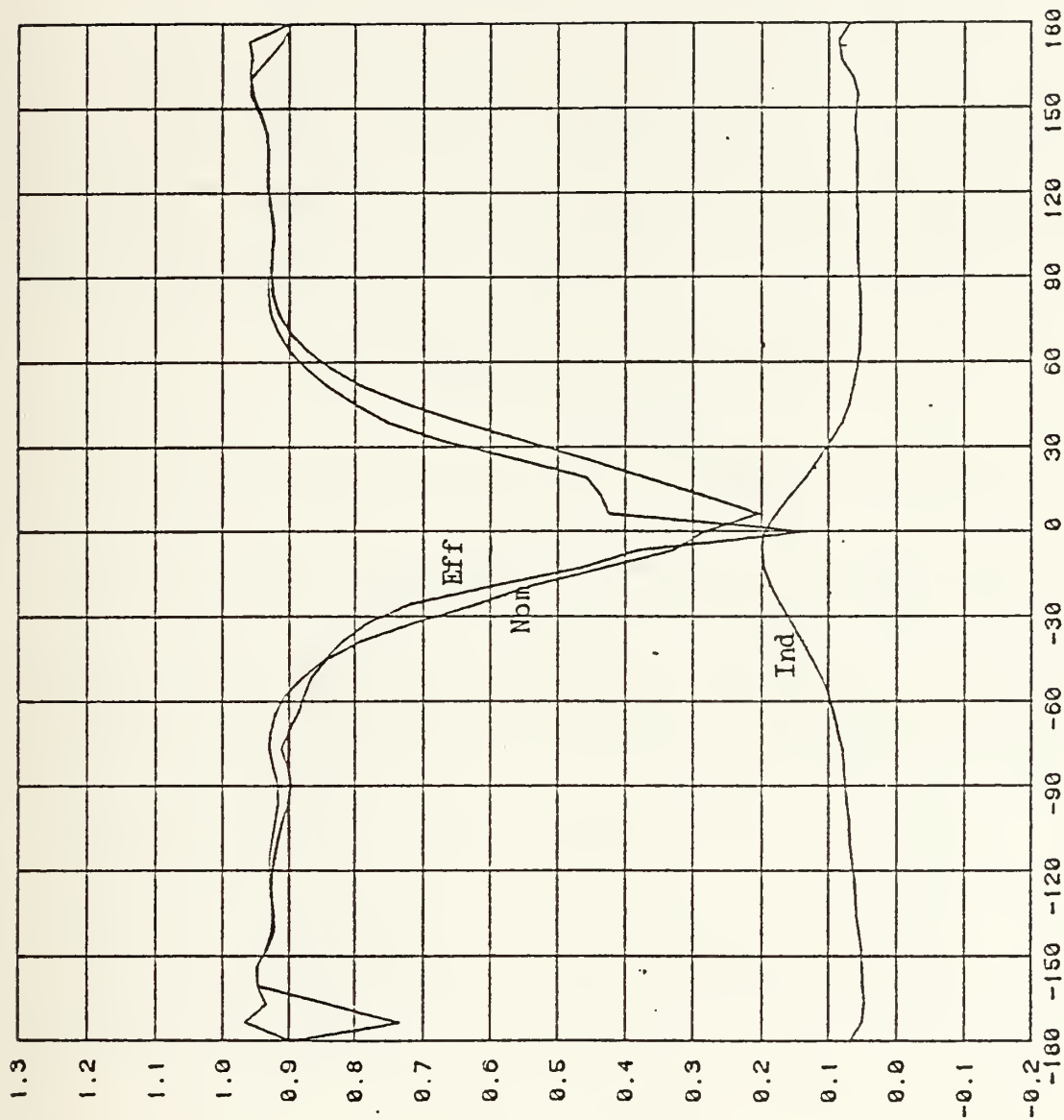


AXIAL WAKE V(R)/V(VOL)

V(VOL)

Nom 1.0
Ind 1.0
Eff 1.0

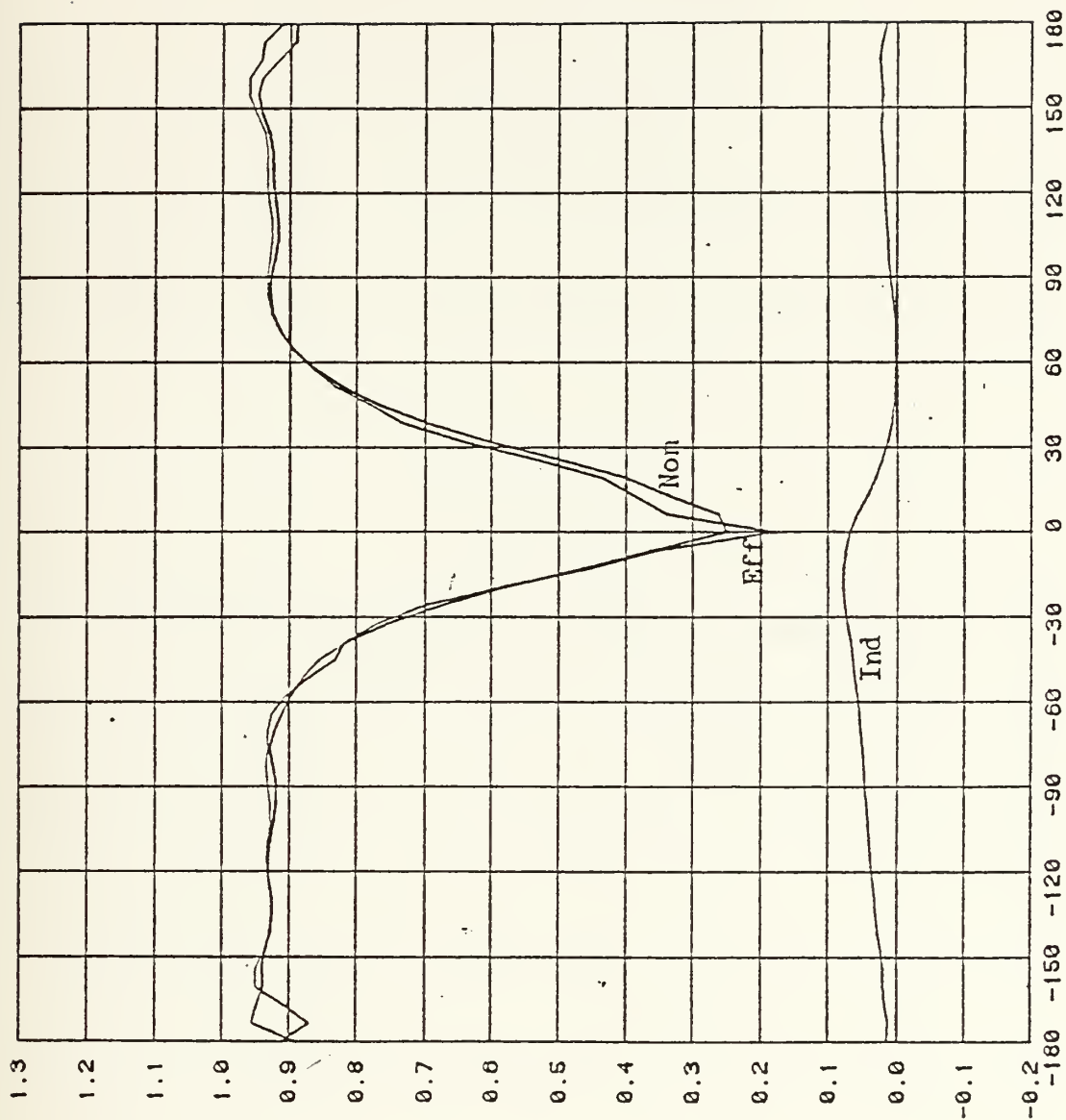
NOM, IND, EFF VELOCITIES, .80 RADIUS



AXIAL WAKE $VCR3/VVOL$

V(VOL)

Nom 1.0
Ind 1.0
Eff 1.0



AXIAL WAKE $V(VOL)$

$V(VOL)$

Nom

Ind

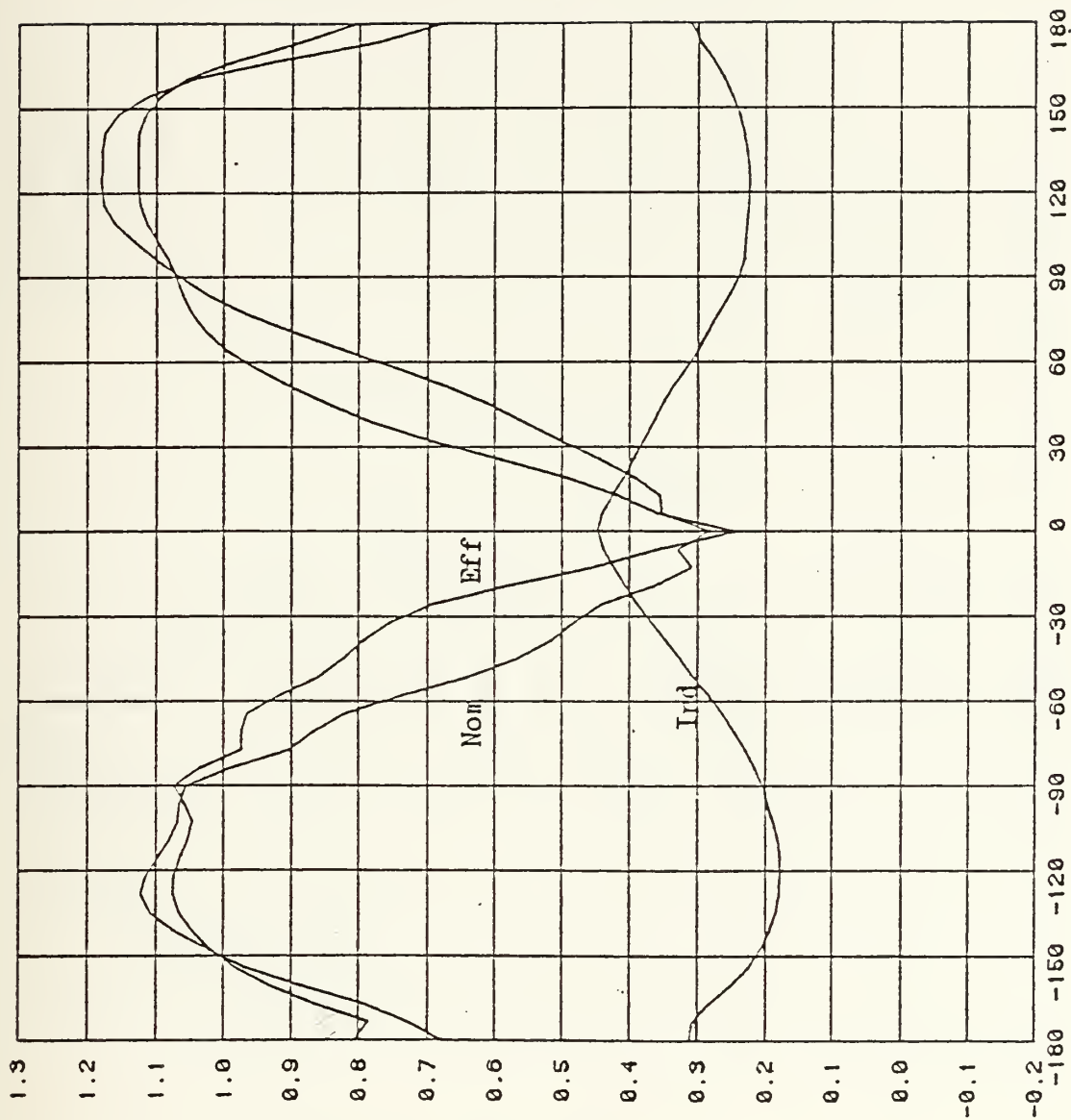
Eff

NOM, IND, EFF VELOCITIES, .98 RADIUS

**** APPENDIX II ****

Nominal, Induced, and Effective Velocities.

Non-Dimensionalized



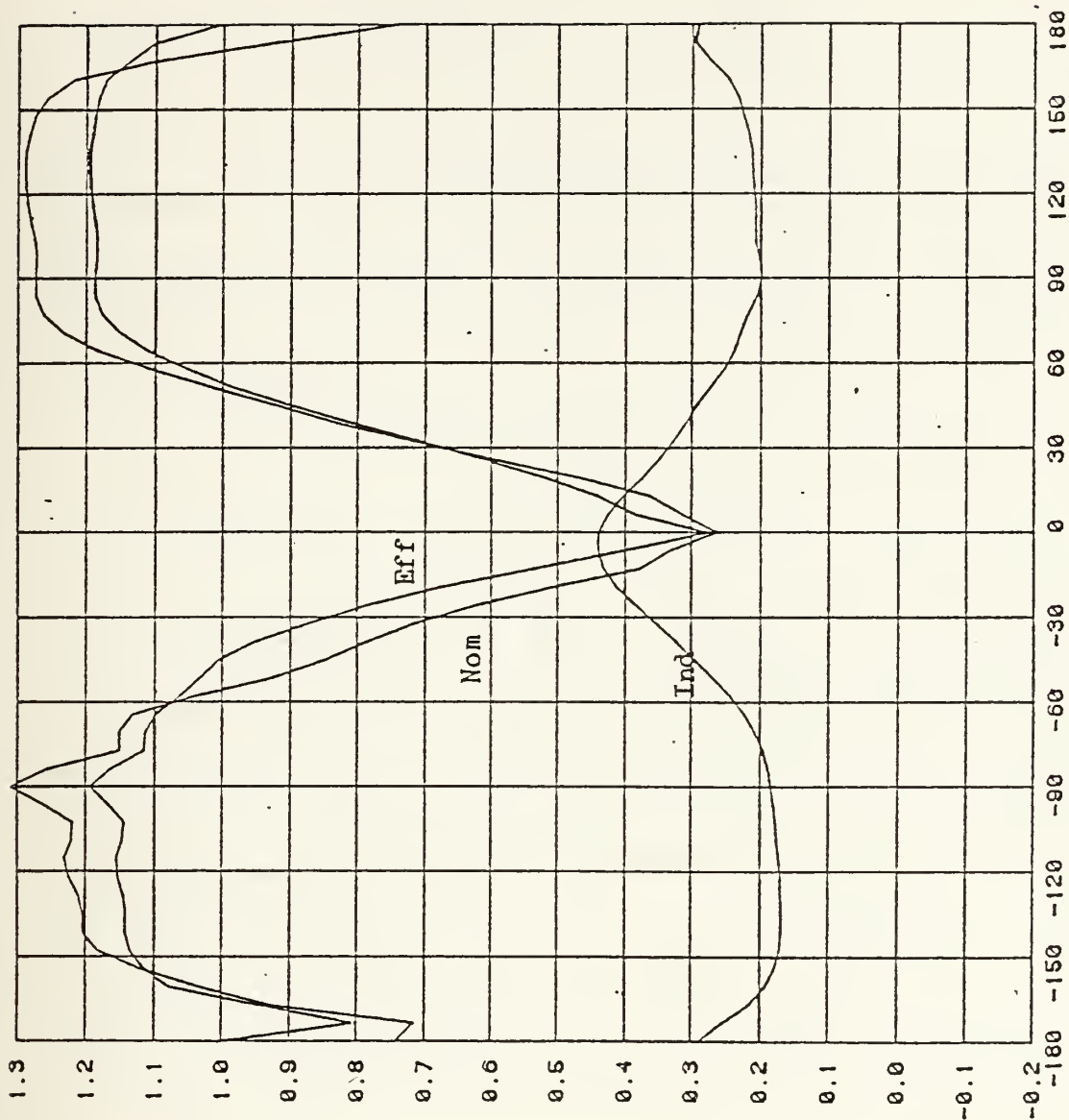
AXIAL WAKE VCR/V(VOL)

V(VOL)
 Nom 0.732
 Ind 1.0
 Eff 0.787

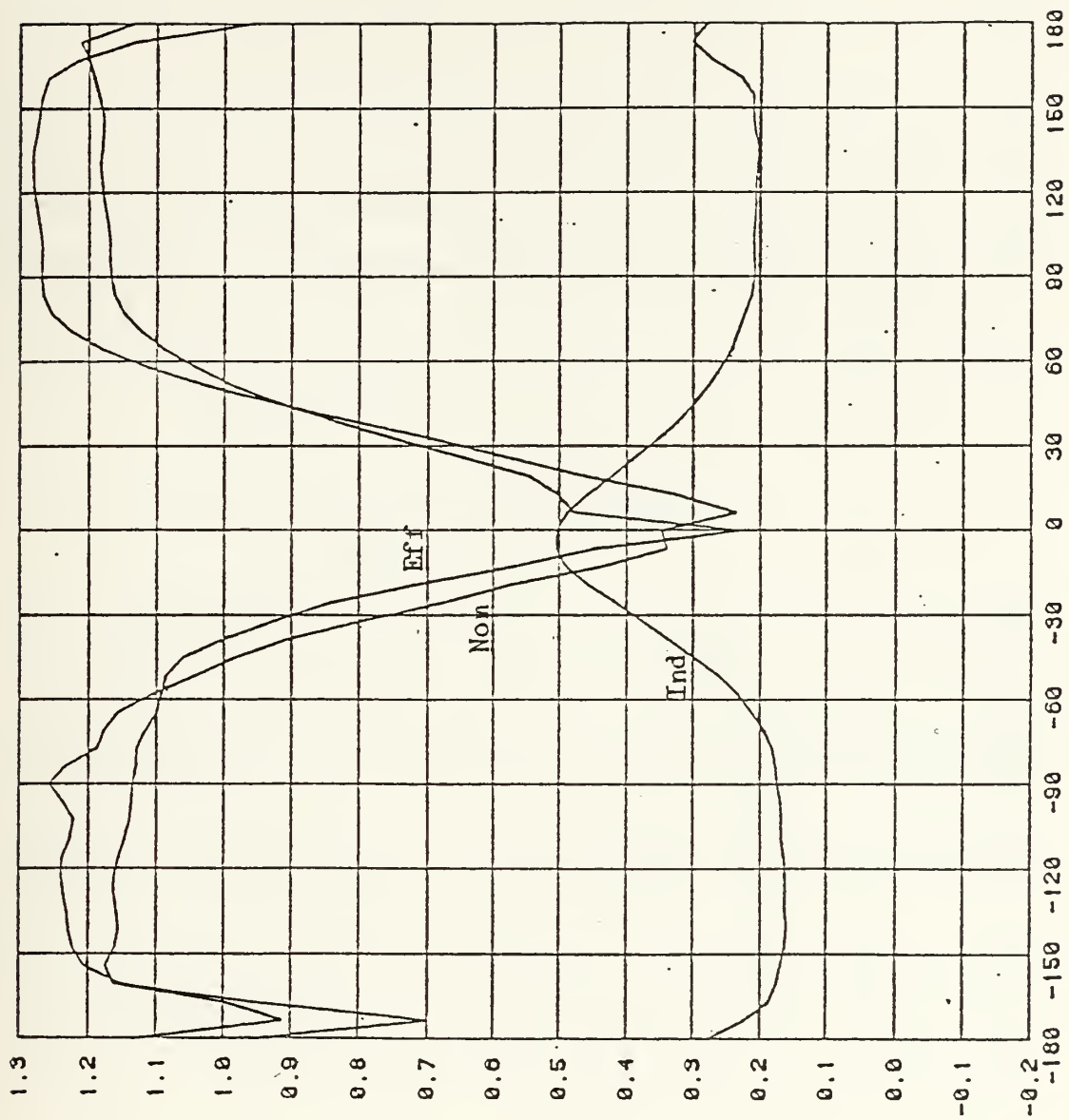
NOM, IND, EFF VELOCITIES, .40 RADIUS

V(VOL)
 Nom 0.732
 Ind 1.0
 Eff 0.787

AXIAL WAKE VCR)/V(VOL)



NOM, IND, EFF VELOCITIES, .60 RADIUS

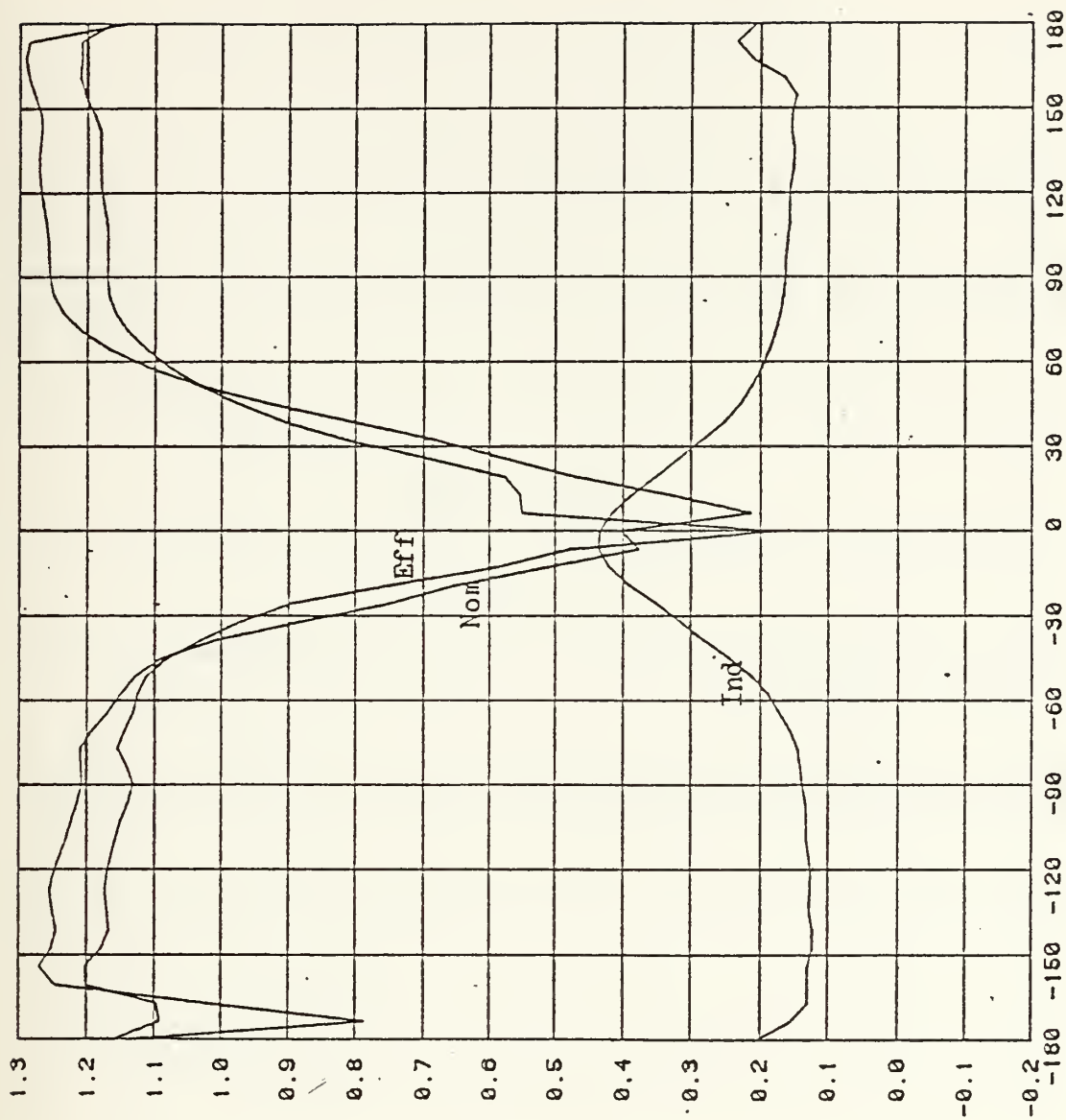


V(VOL)
 Nom 0.732
 Ind 1.0
 Eff 0.787

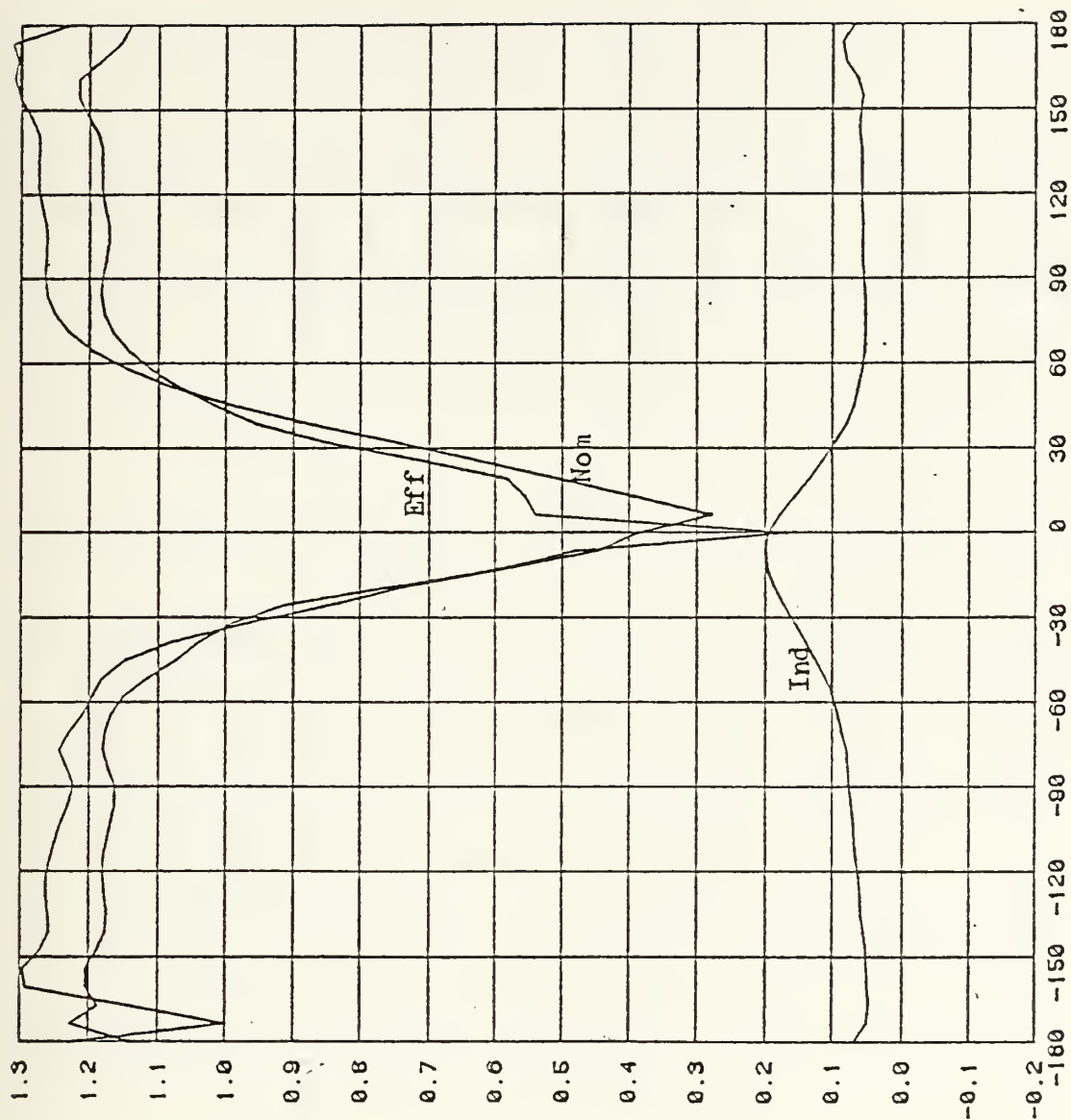
NOM, IND, EFF VELOCITIES, .70 RADIUS

V(VOL)
 Nom 0.732
 Ind 1.0
 Eff 0.787

AXIAL WAKE VCR3/V(VOL)



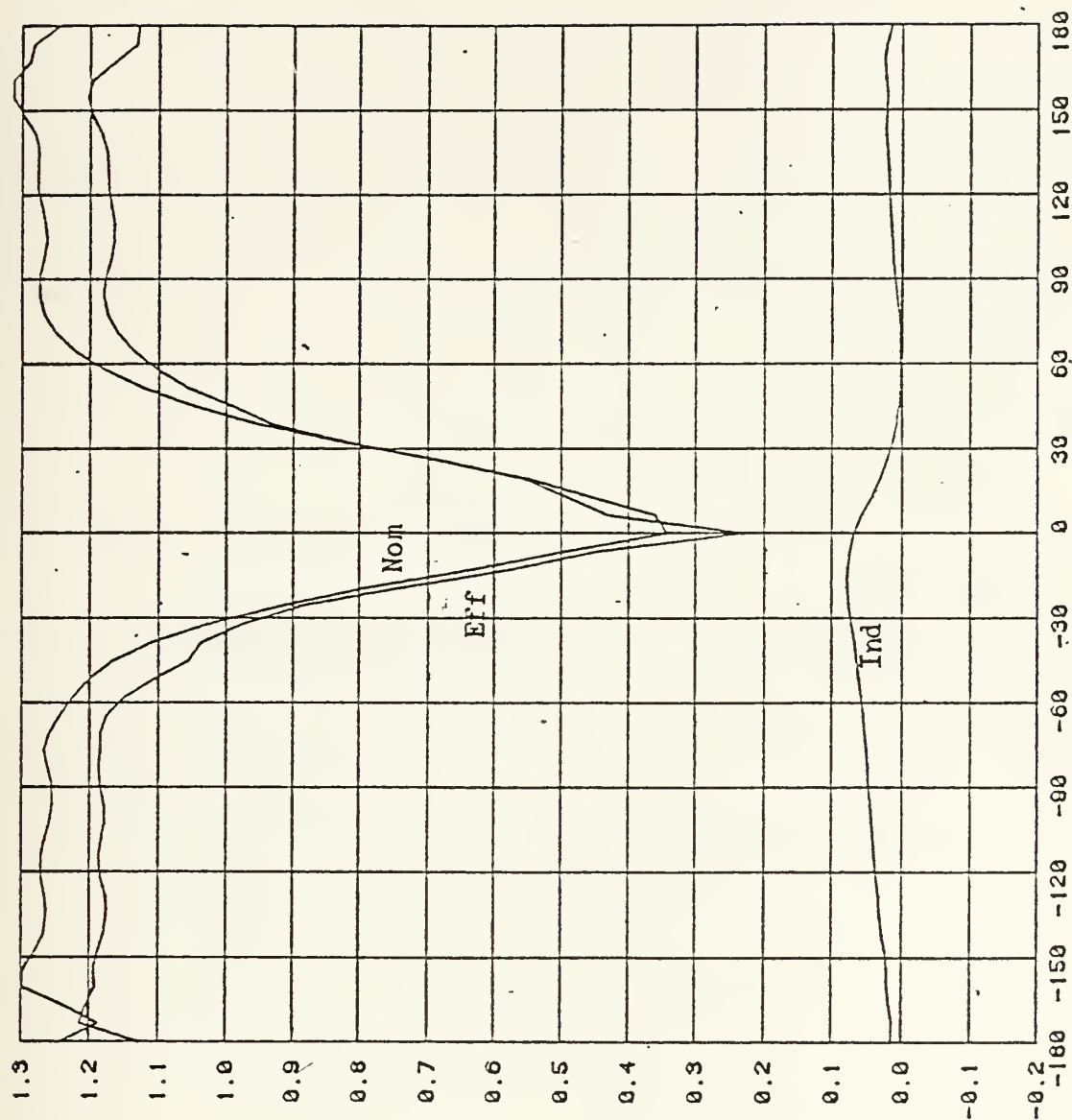
NOM, IND, EFF VELOCITIES, .80 RADIUS



AXIAL WAKE V(CR)/V(VOL)

V(VOL)
 Nom 0.732
 Ind 1.0
 Eff 0.787

NOM, IND, EFF VELOCITIES, .90 RADIUS



V(VOL)
Nom 0.732
Ind 1.0
Eff 0.787

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

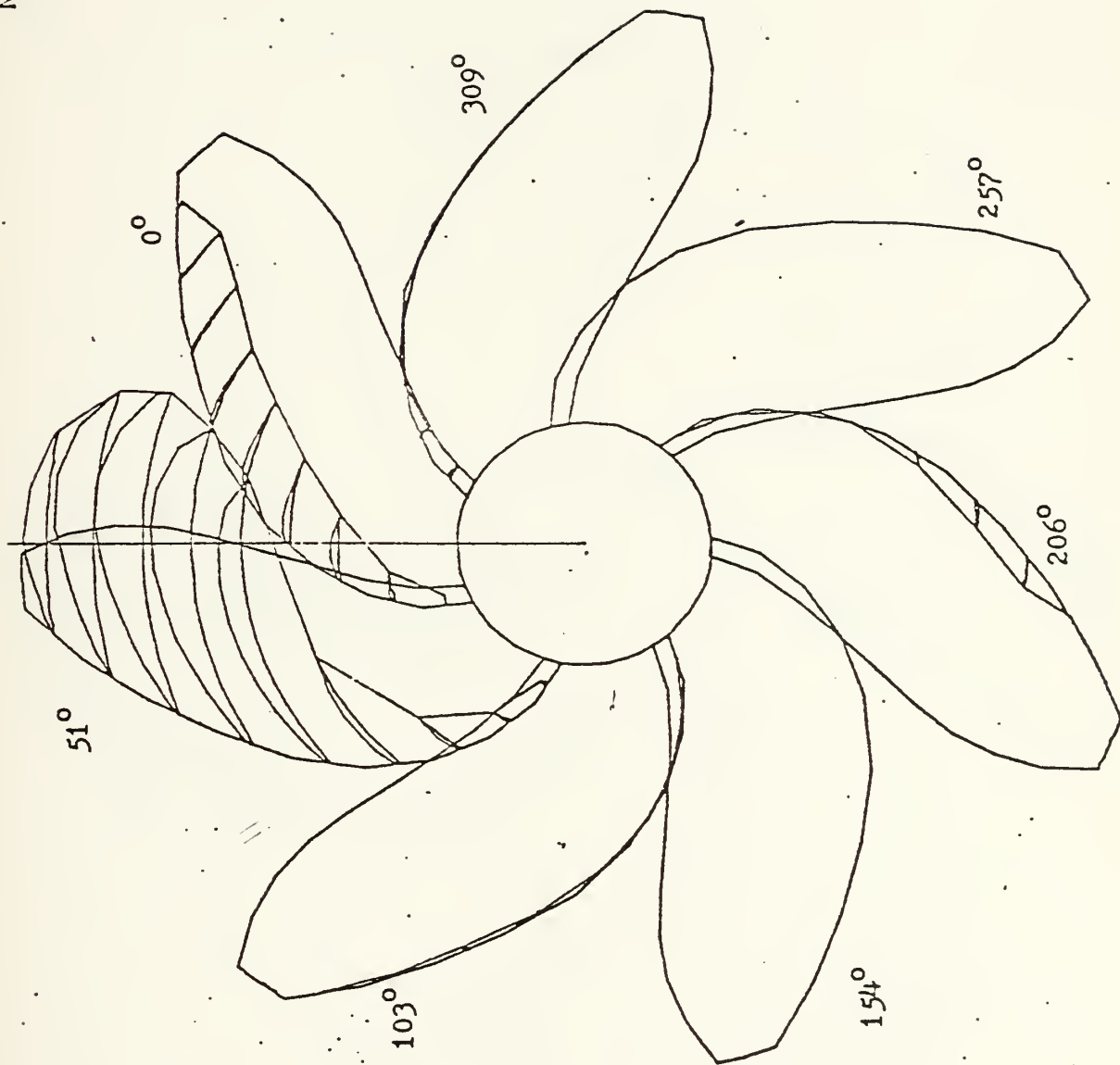
1493

1494

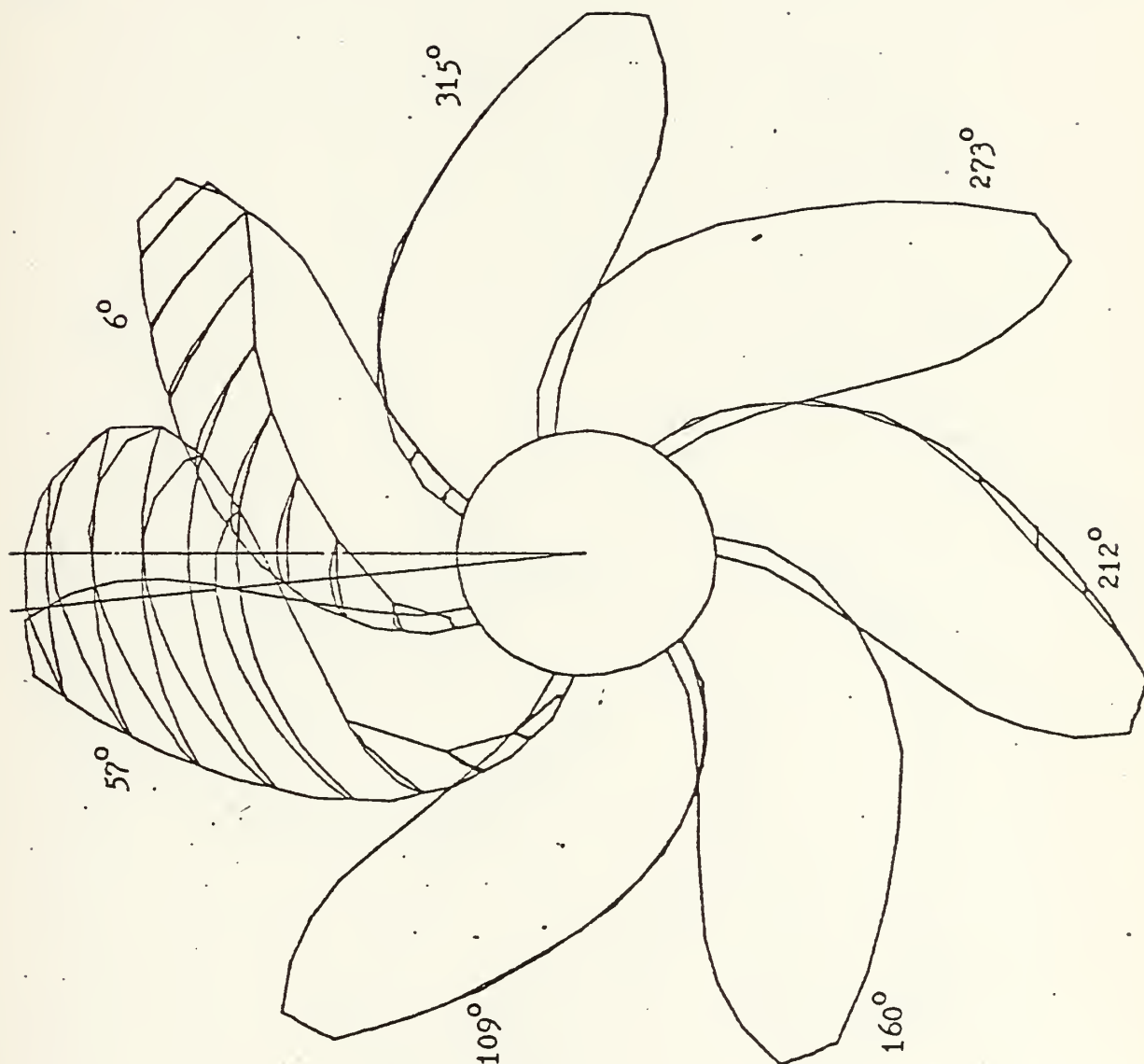
1495

1496

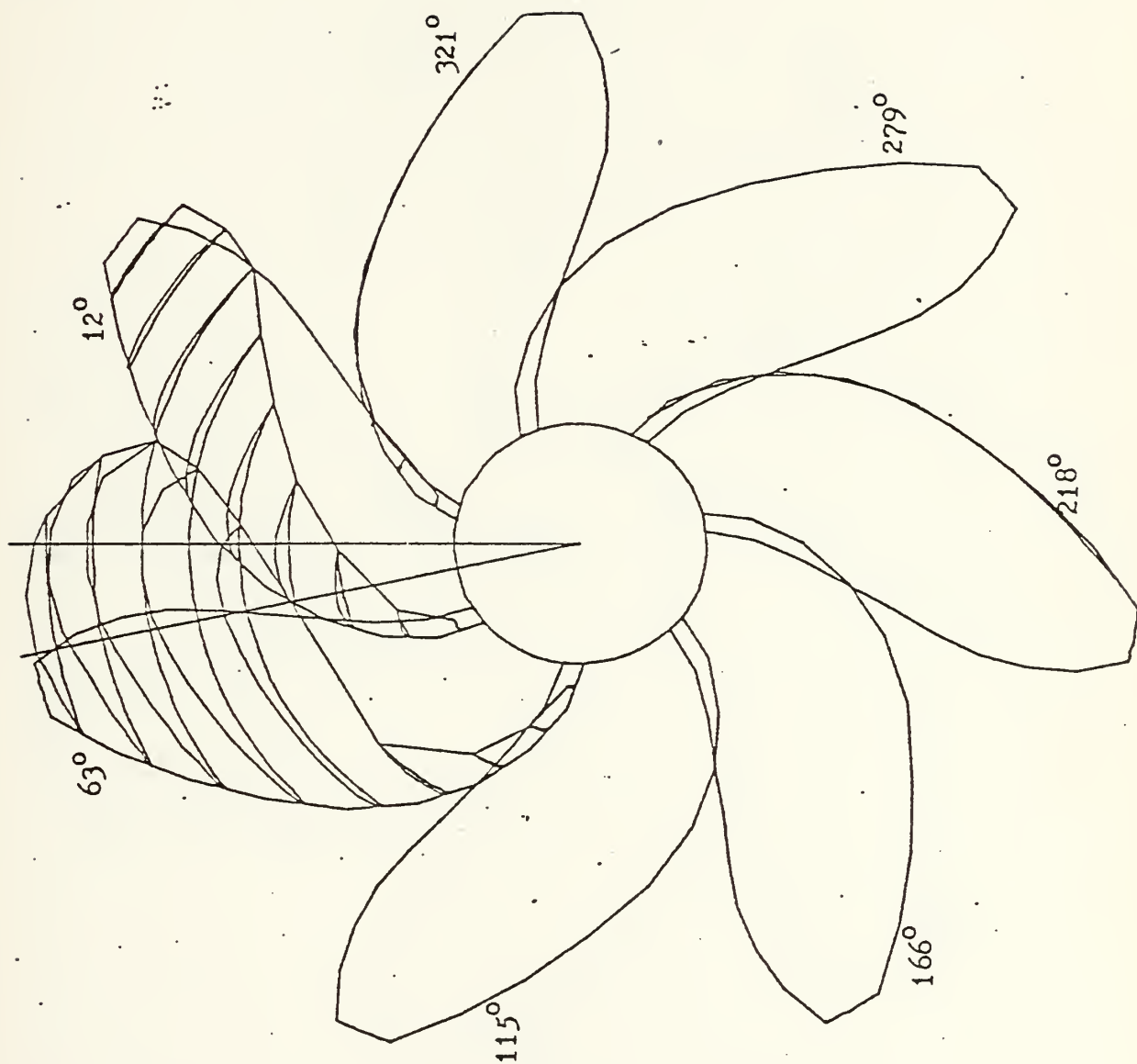
Nominal Wake
.0



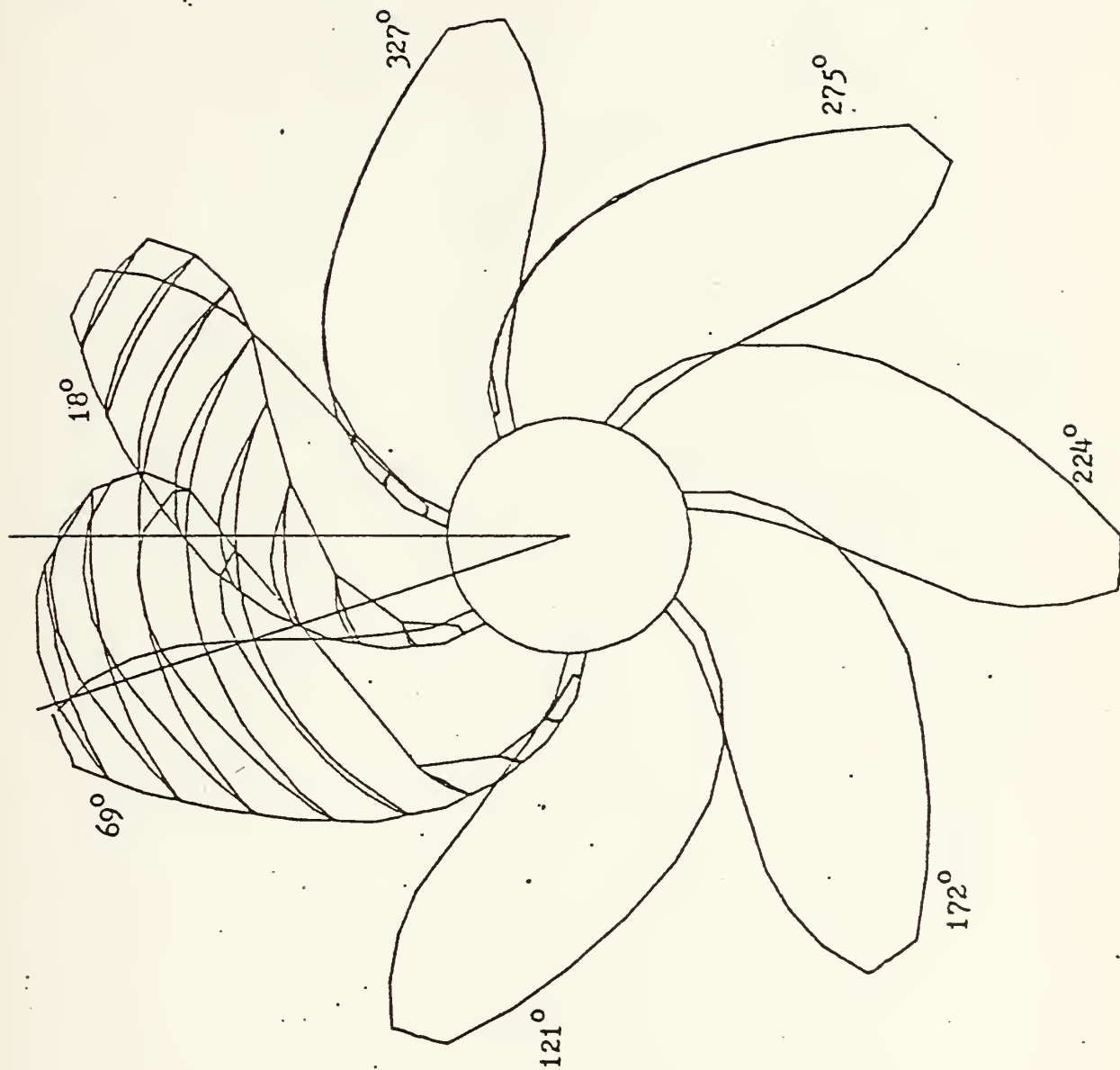
Nominal Wake
6°



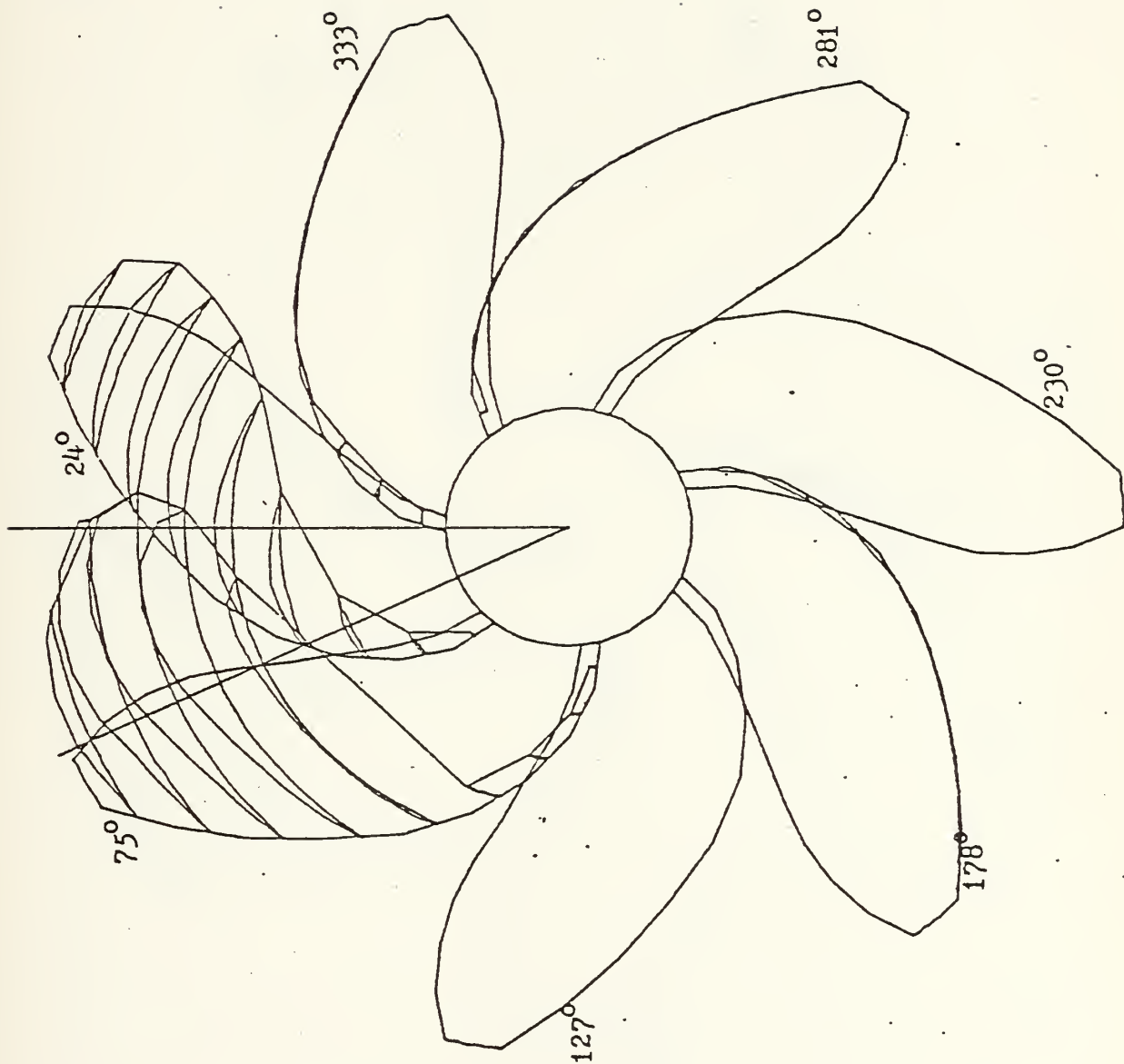
Nominal Wake
12°



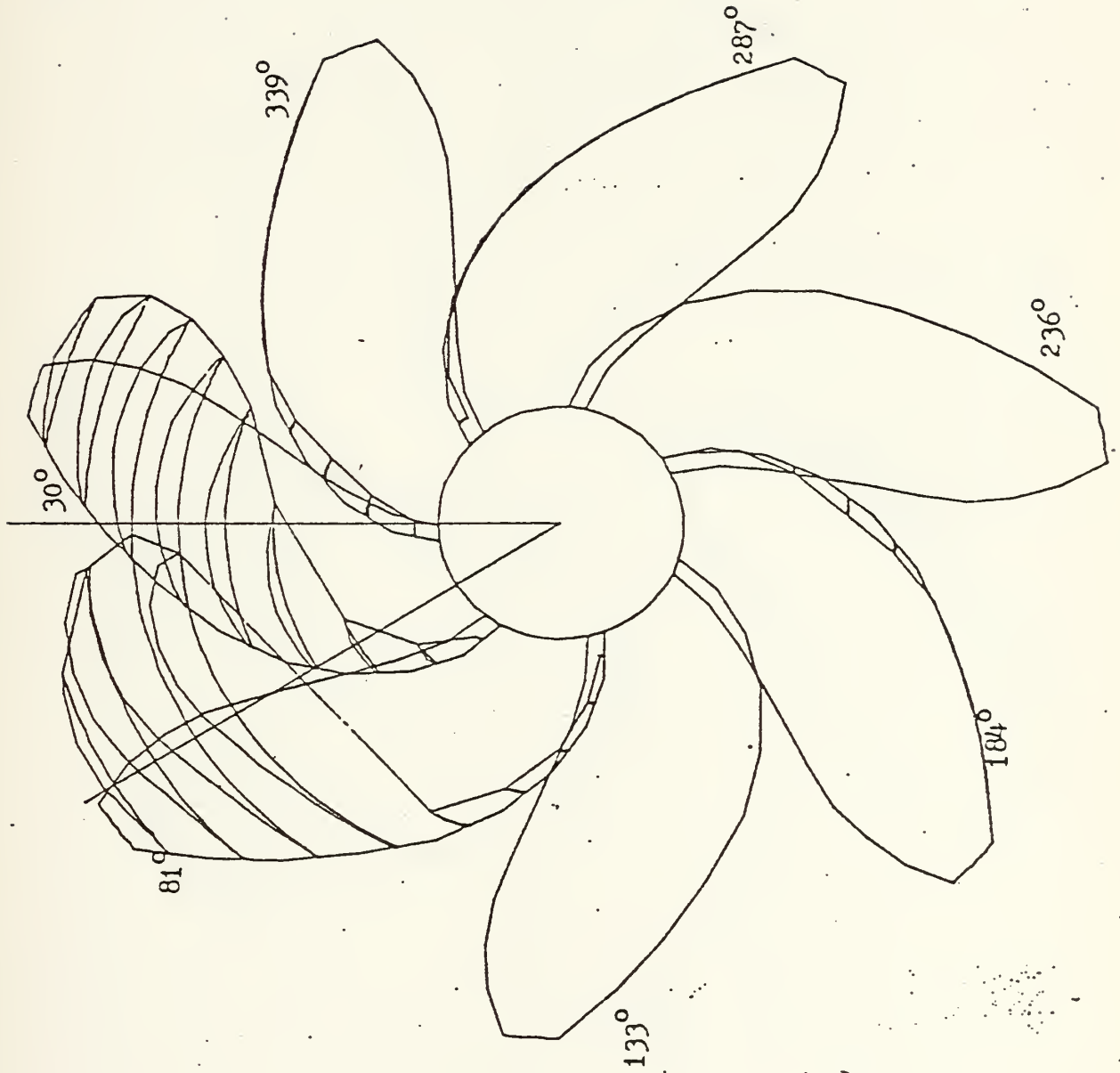
Nominal Wake
18°



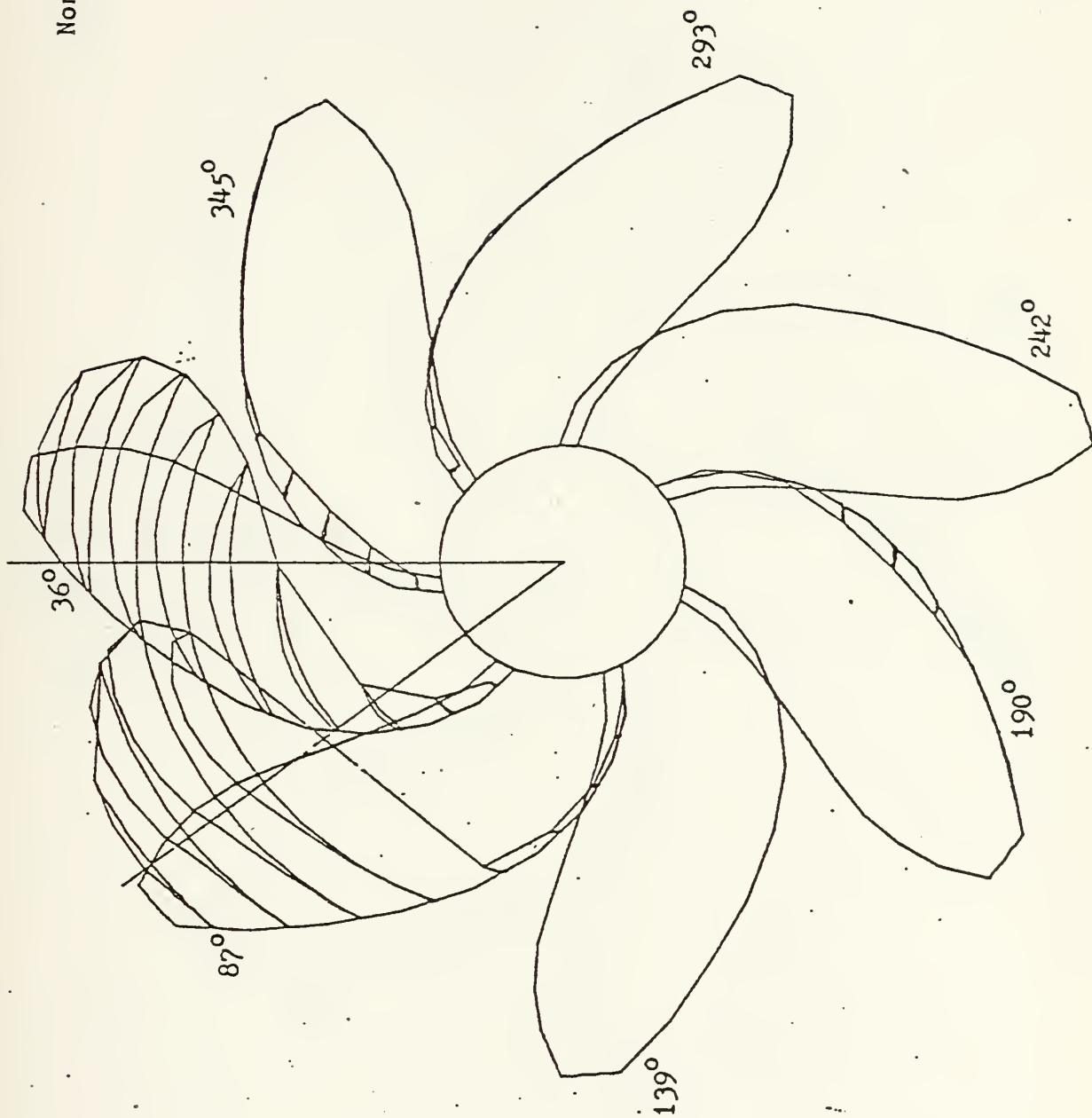
Nominal Wake
24°



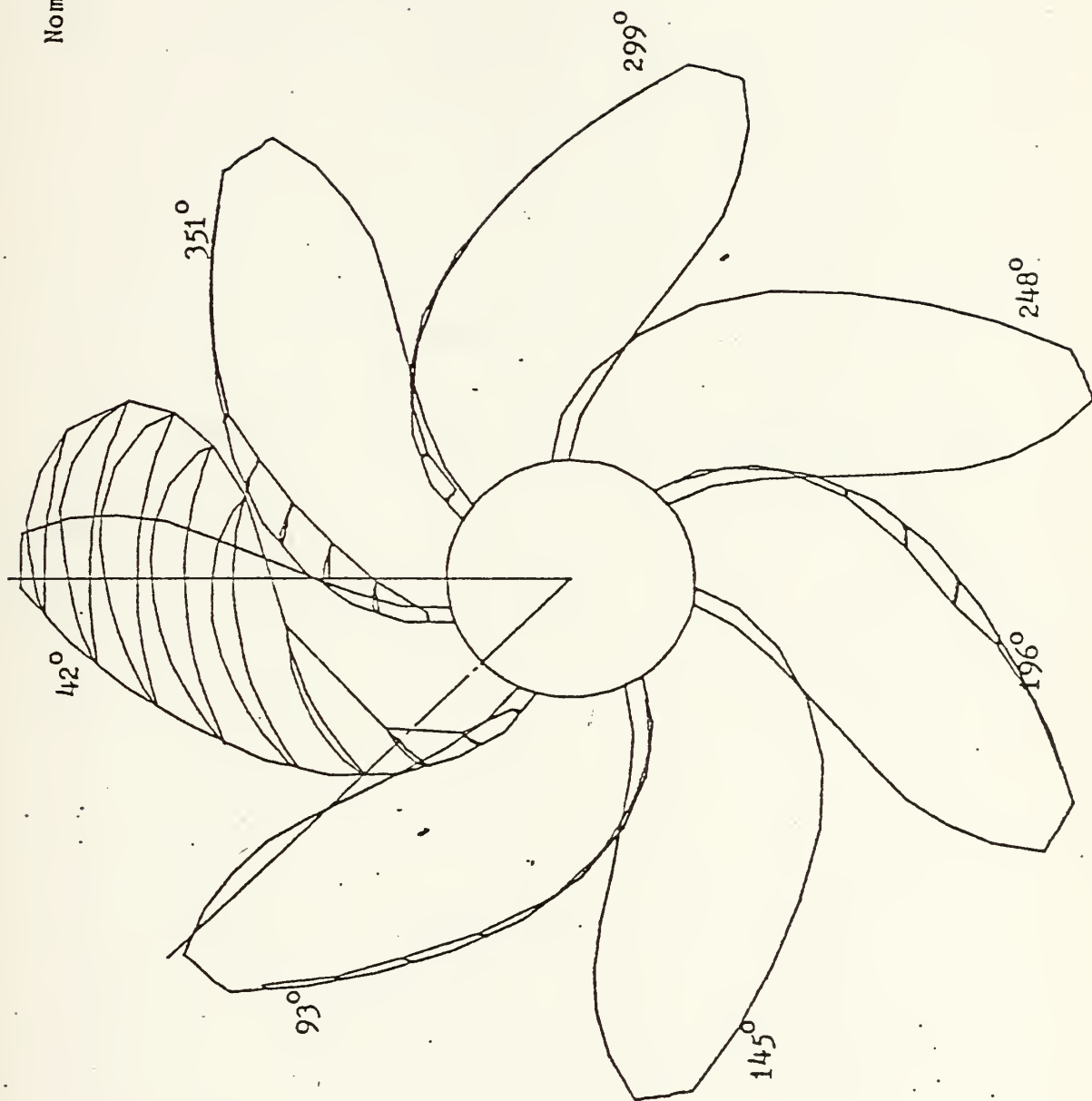
Nominal Wake
30°



Nominal Wake
36°



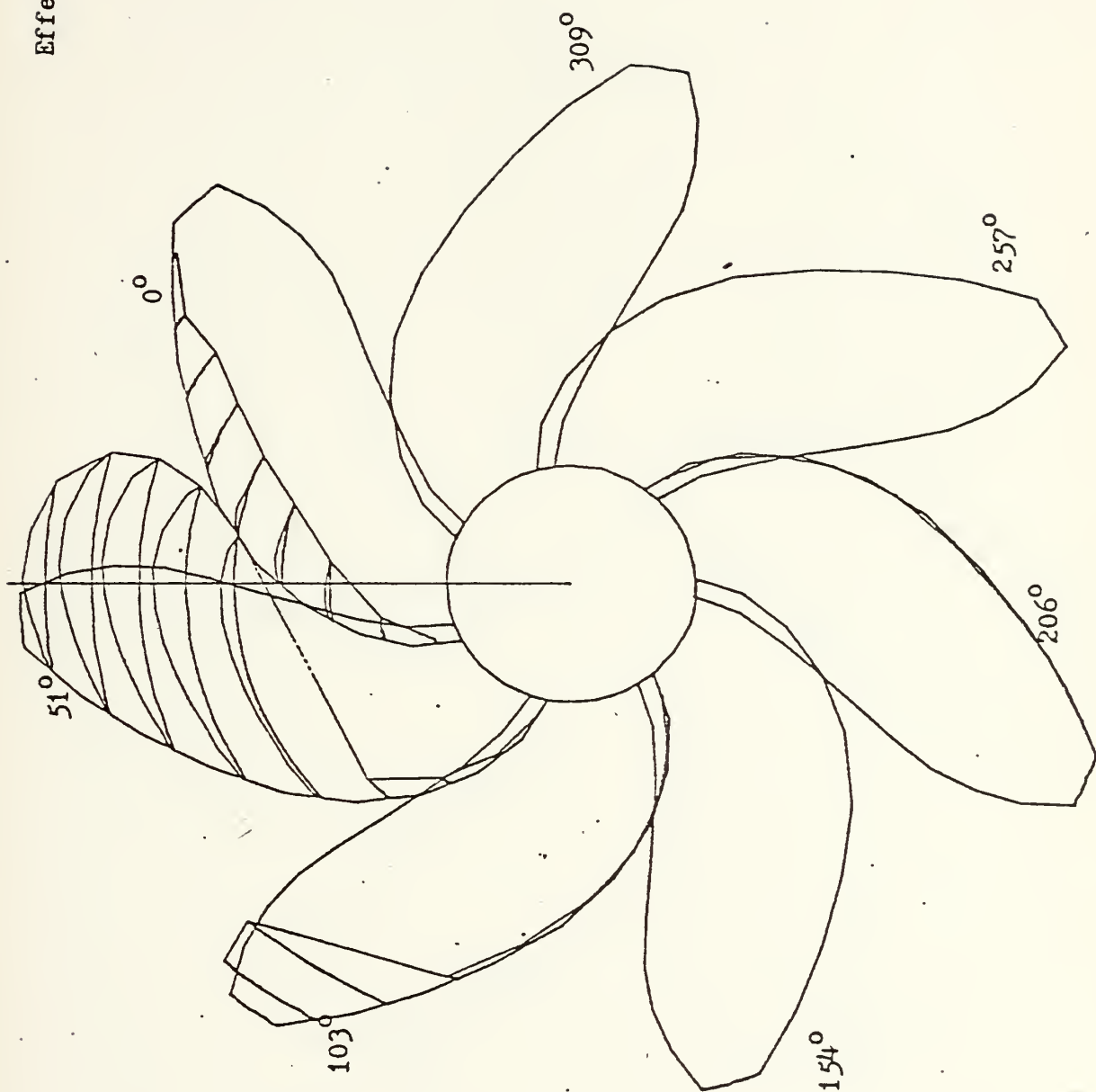
Nominal Wake
42°



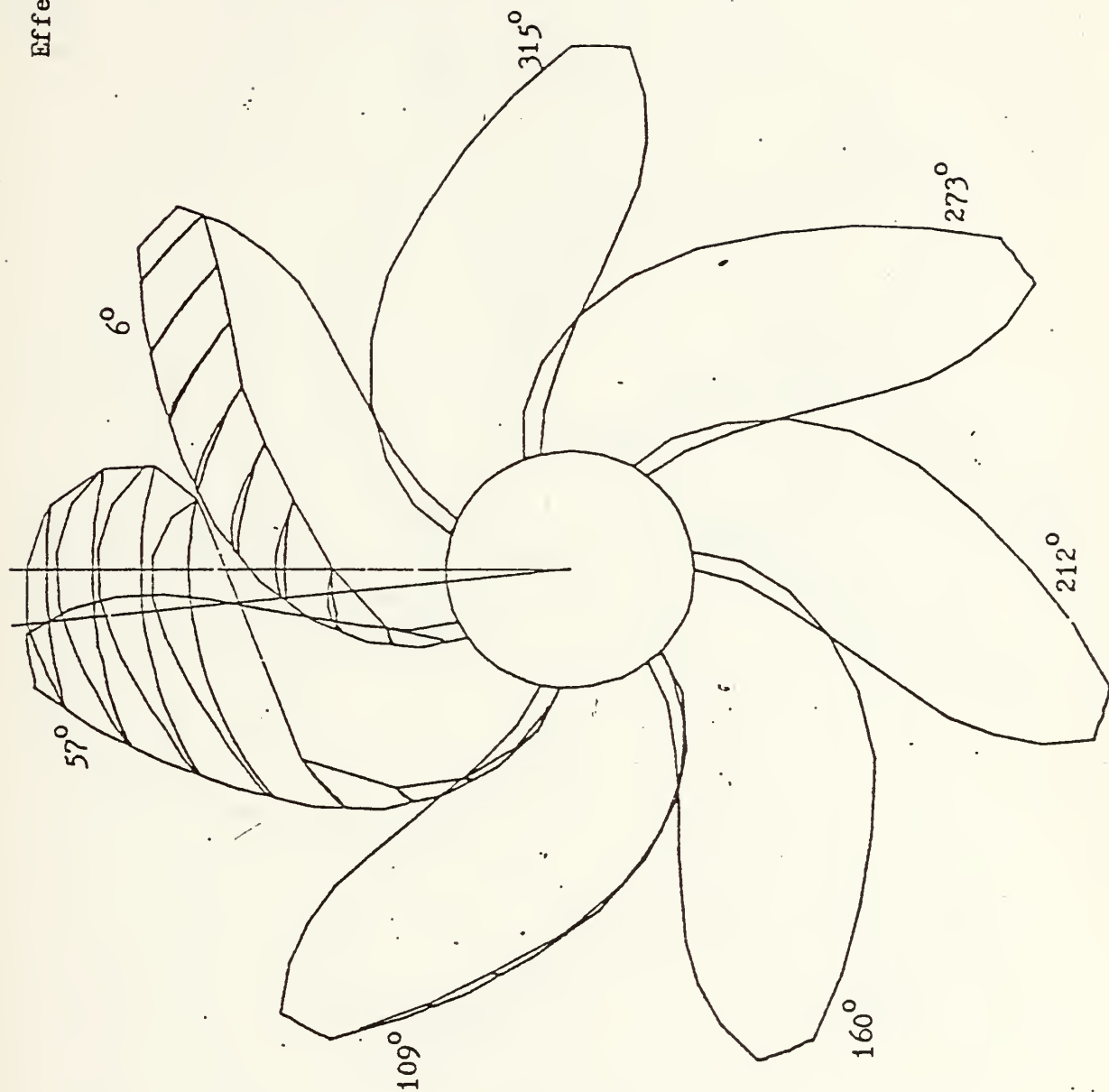
**** APPENDIX IV ****

Propeller Cavitation, Effective Wake

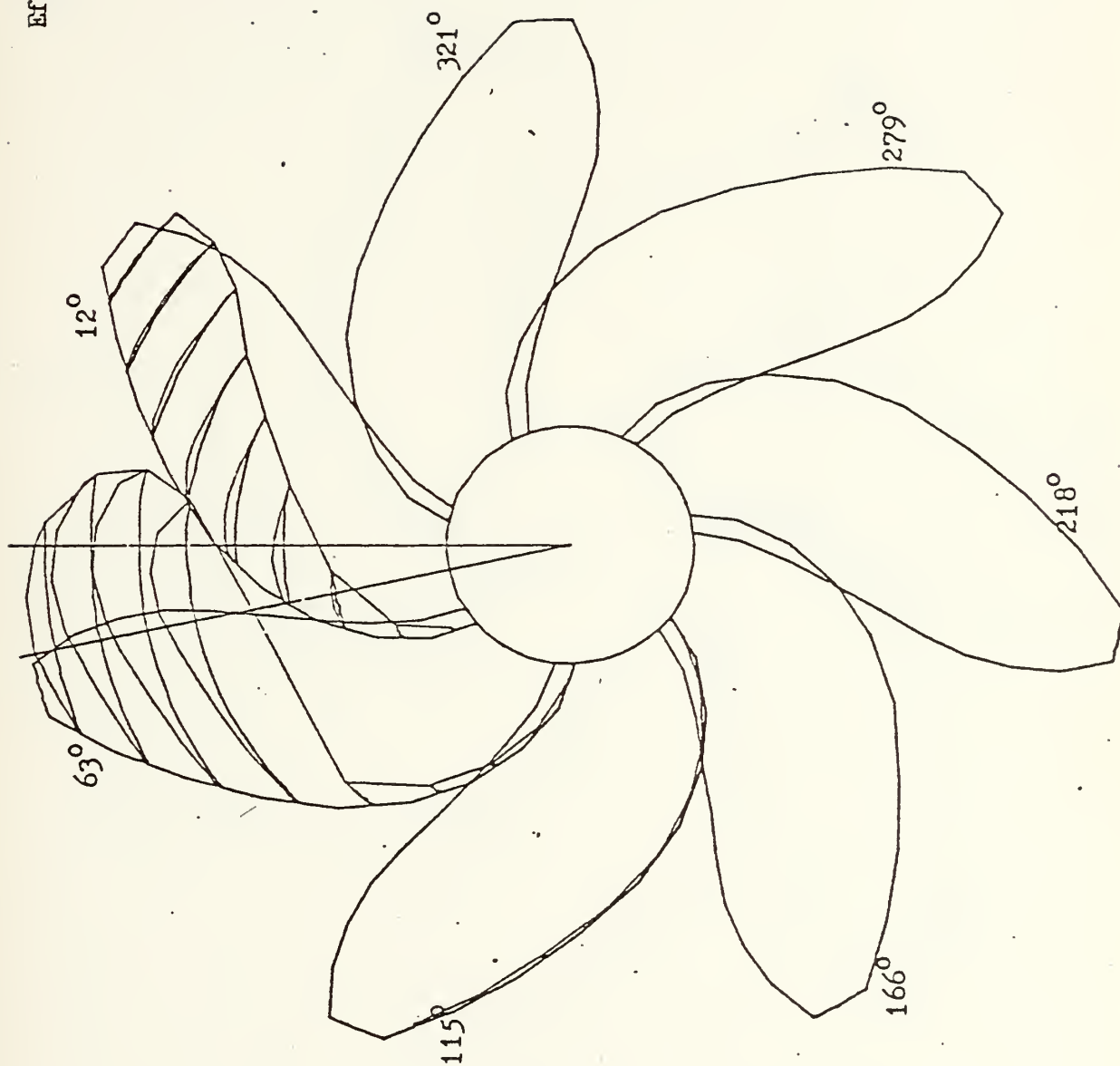
Effective Wake
0°



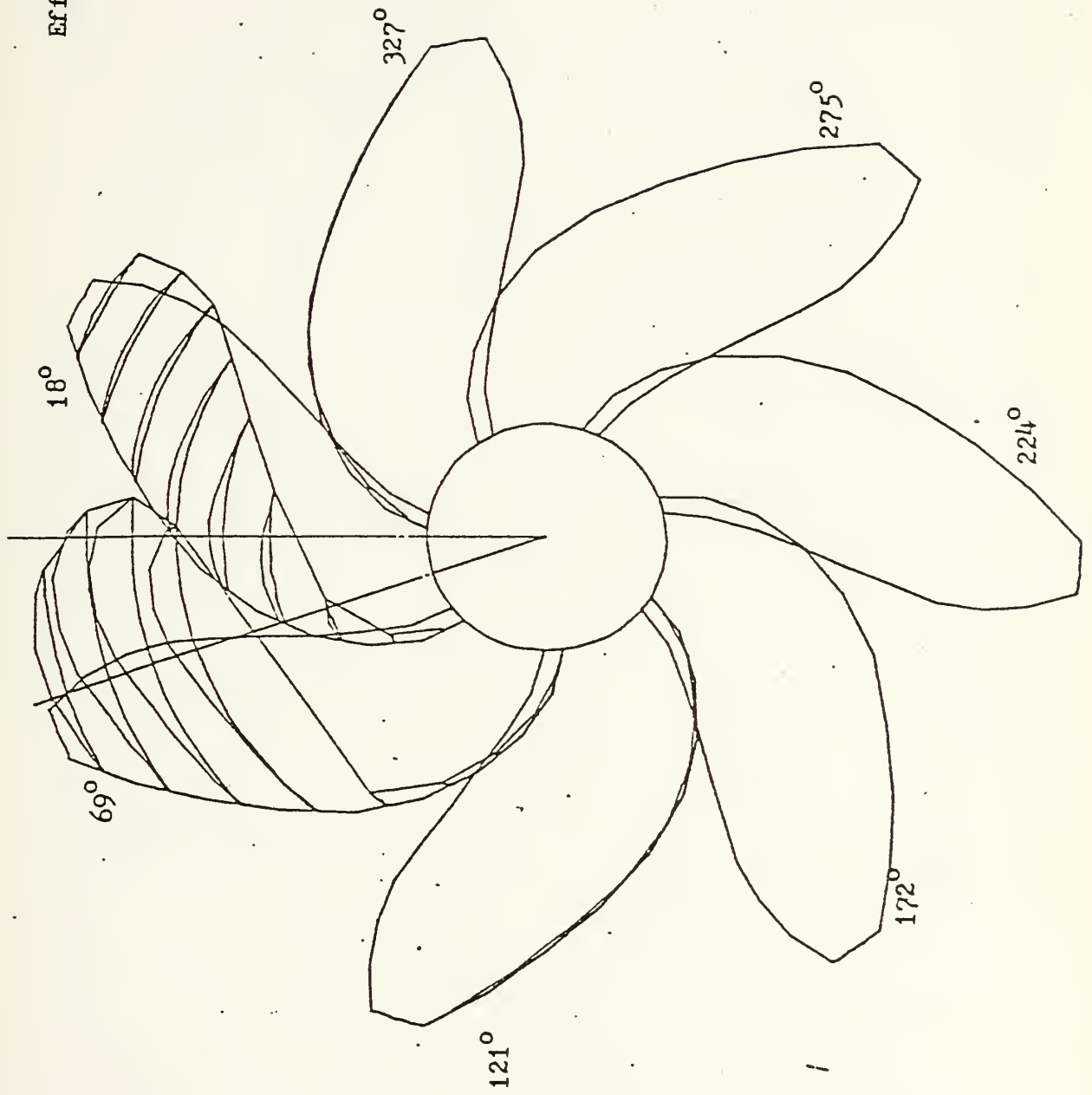
Effective Wake
6°



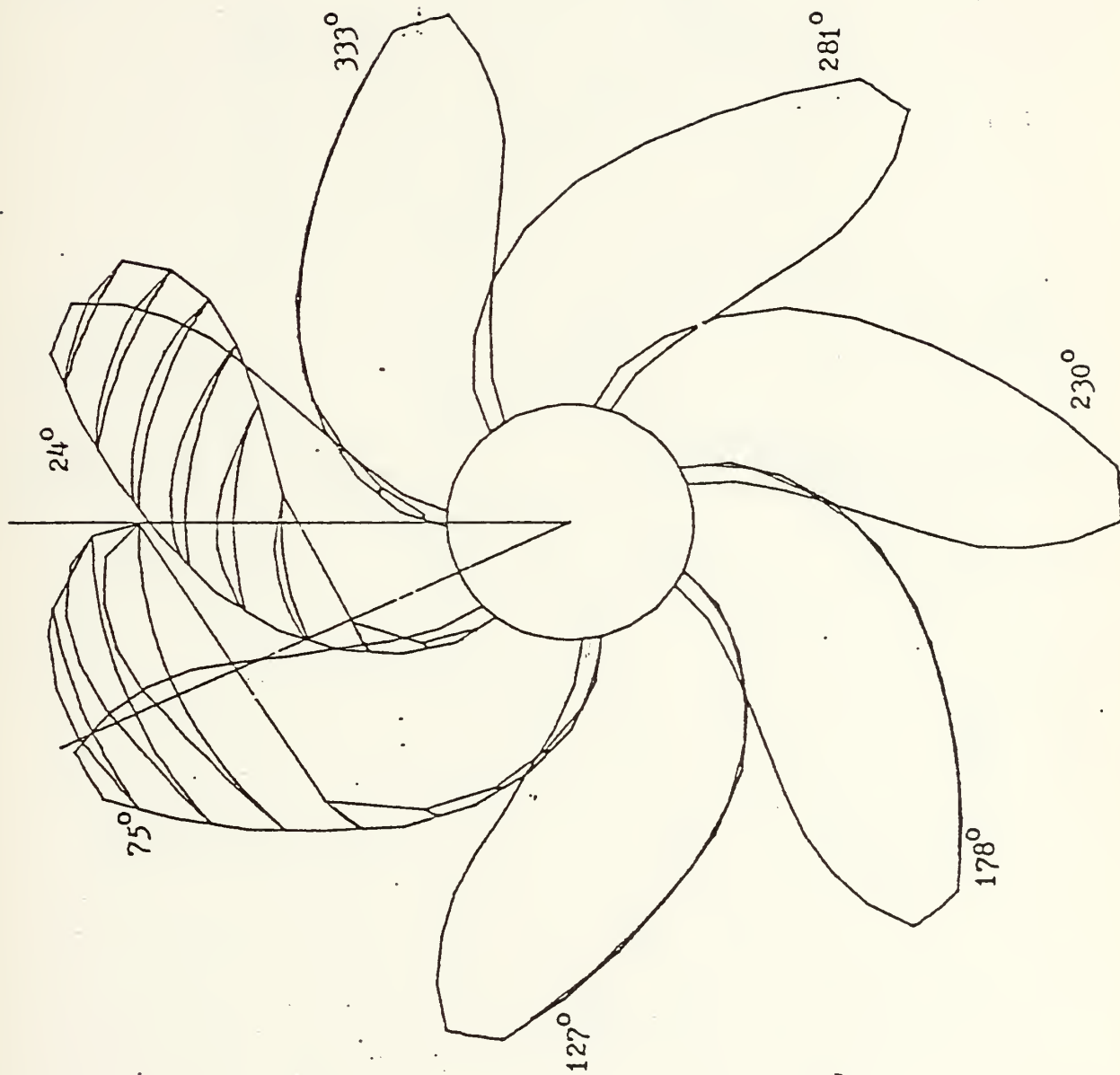
Effective Wake
12°



Effective Wake
18°

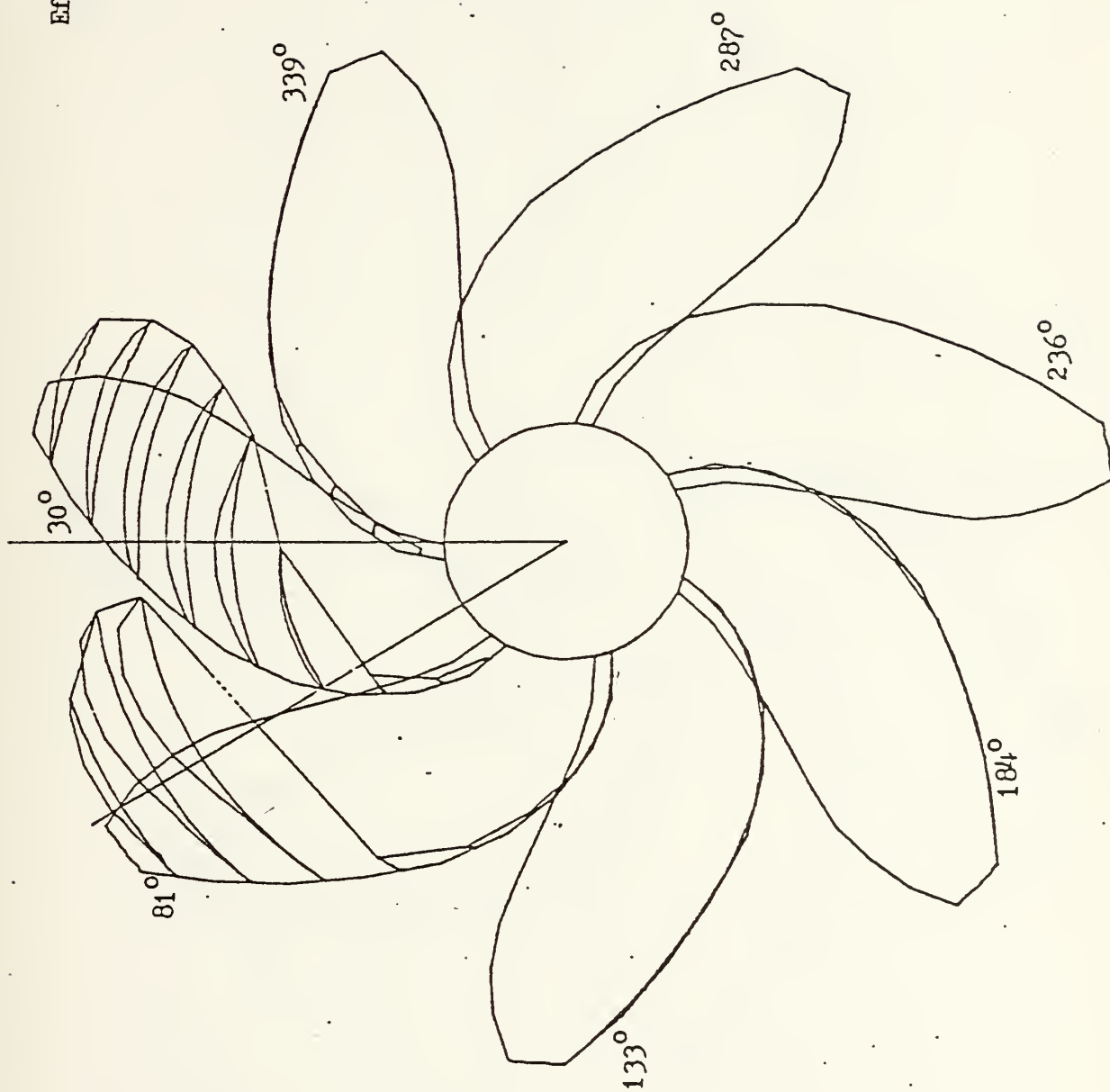


Effective Wake
24°

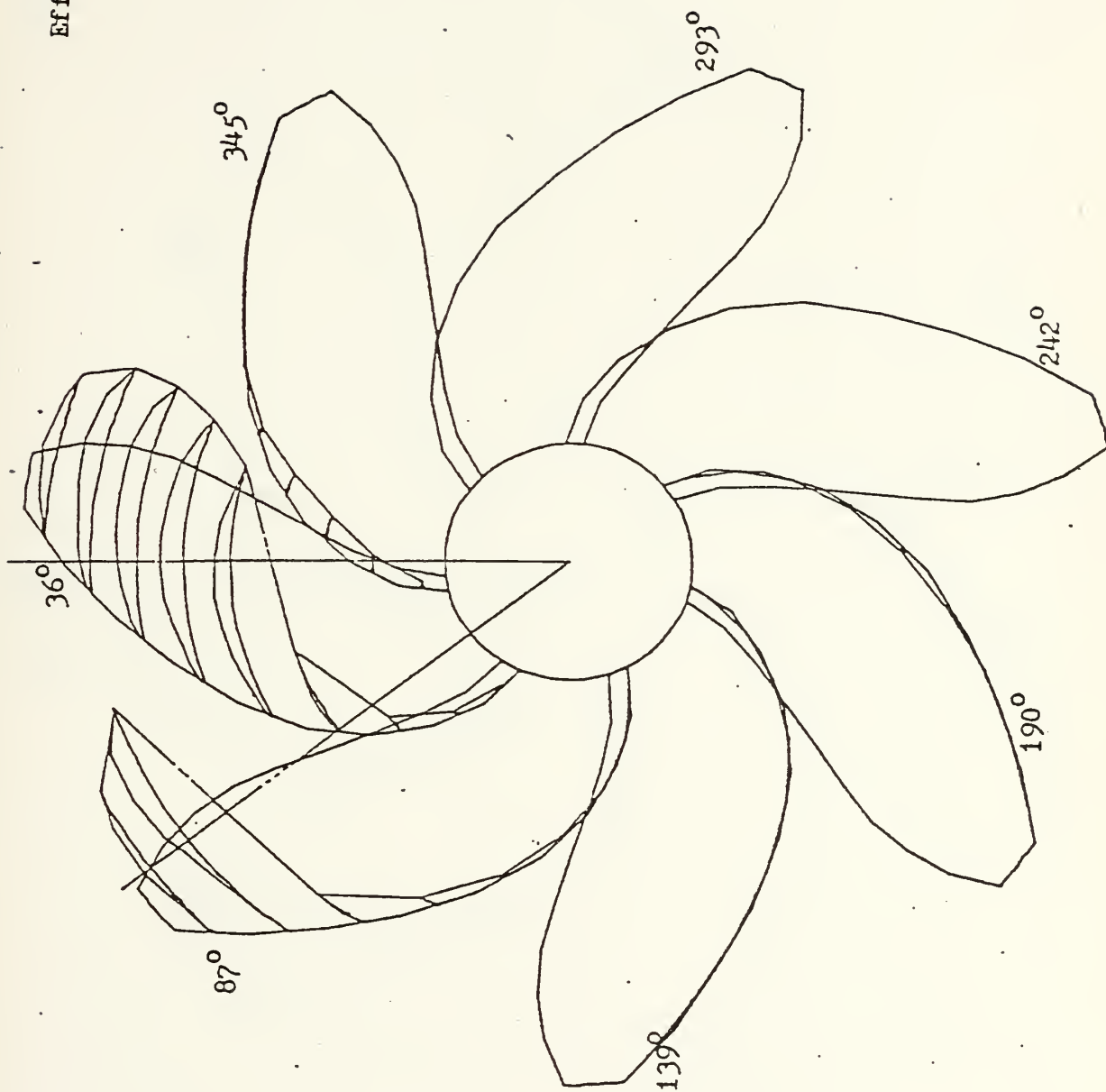


Effective Wake
30°

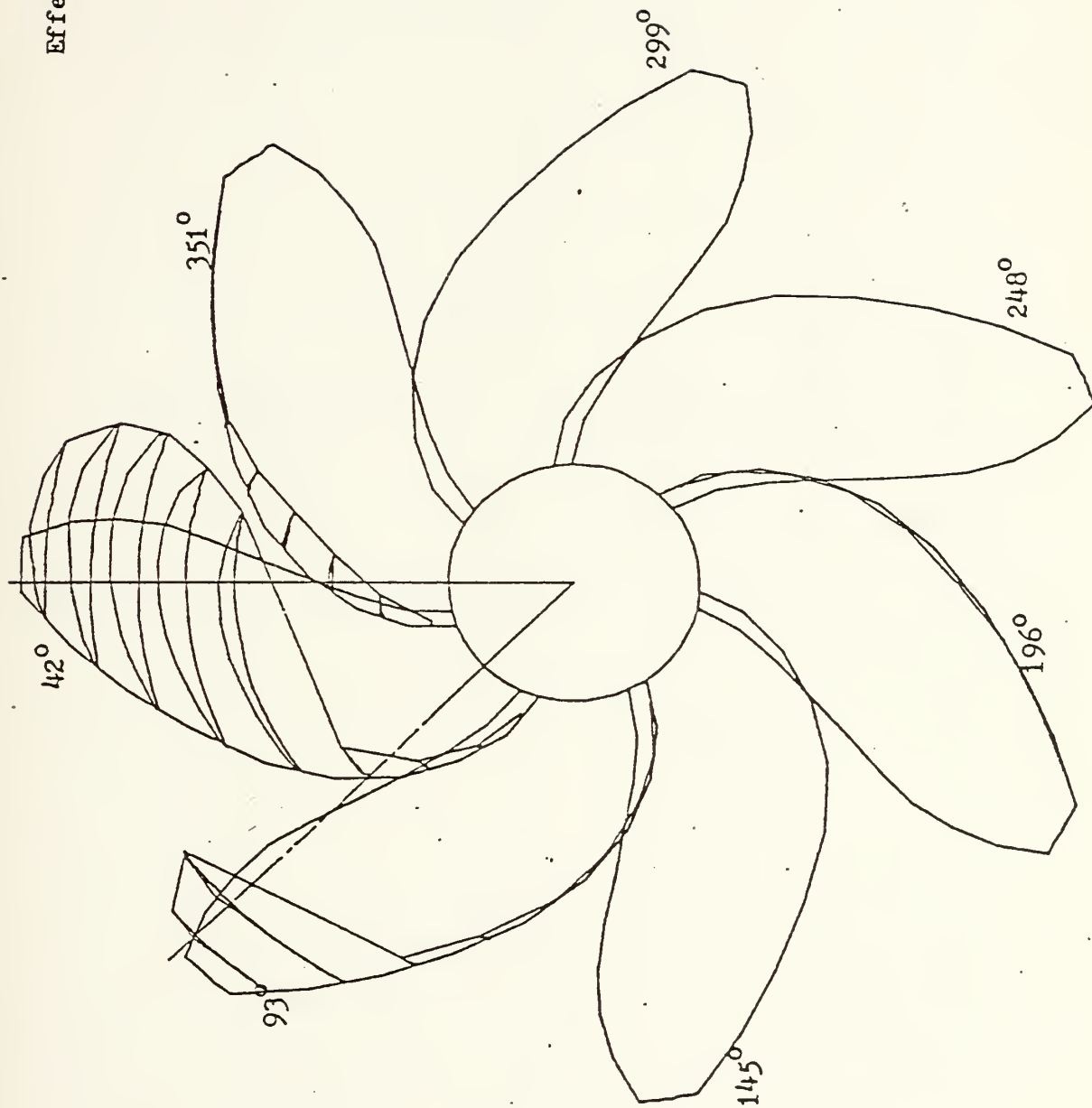
W



Effective Wake
36°



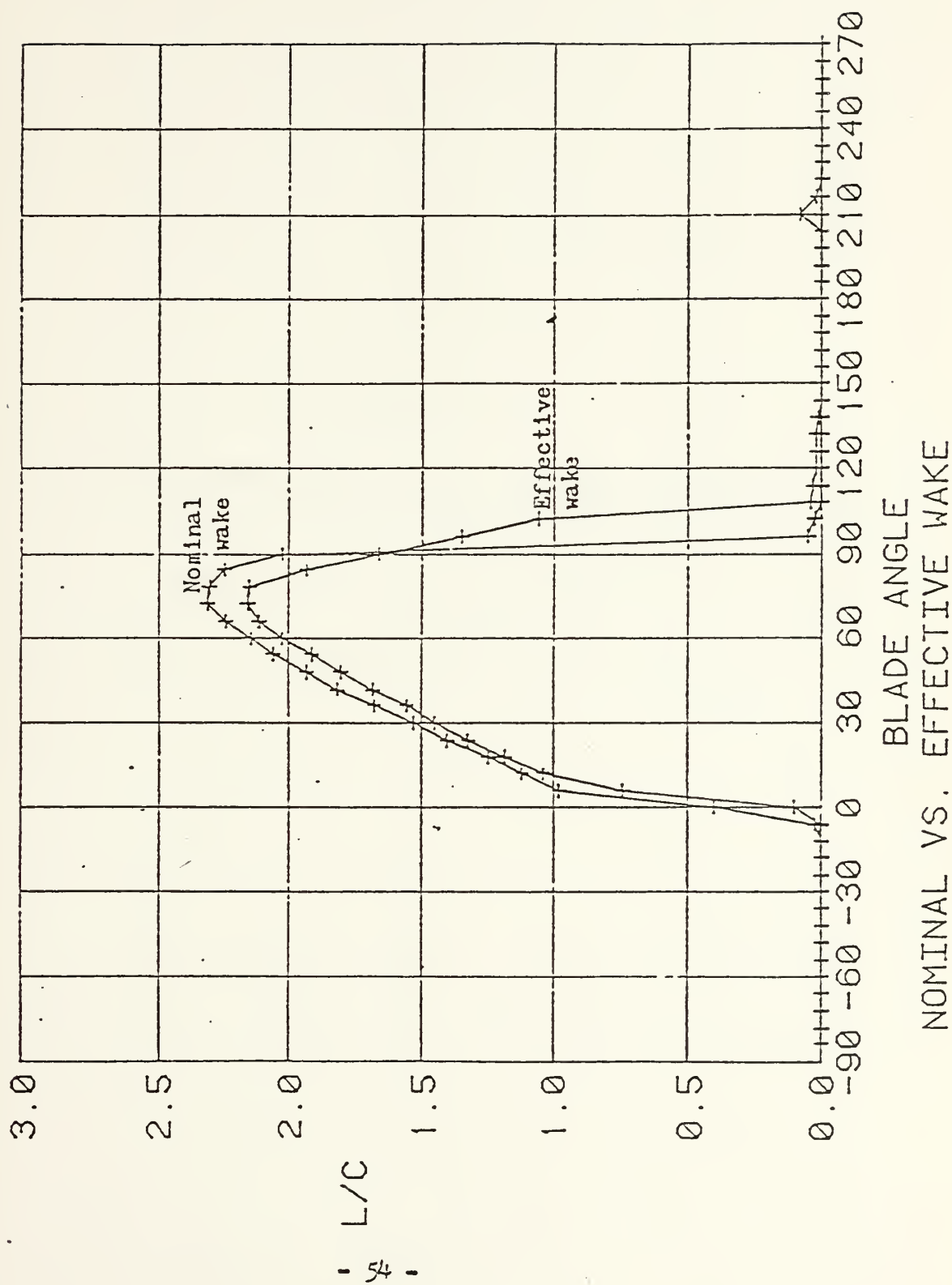
Effective Wake
42°



** APPENDIX V **

Cavity Length, Nominal and Effective Wakes

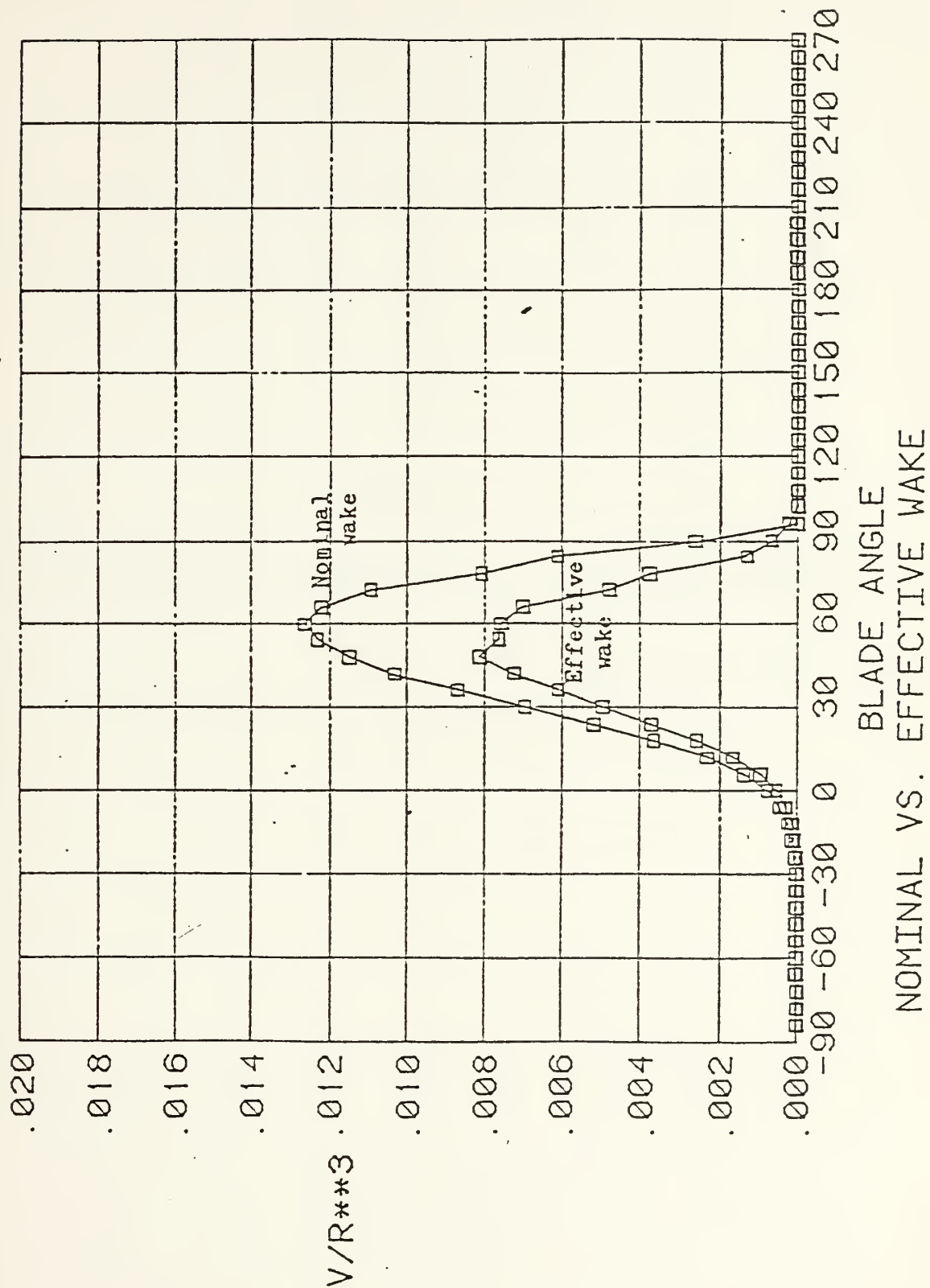
PUF-3 CAVITY LENGTH OUTPUT



** APENDIX VI **

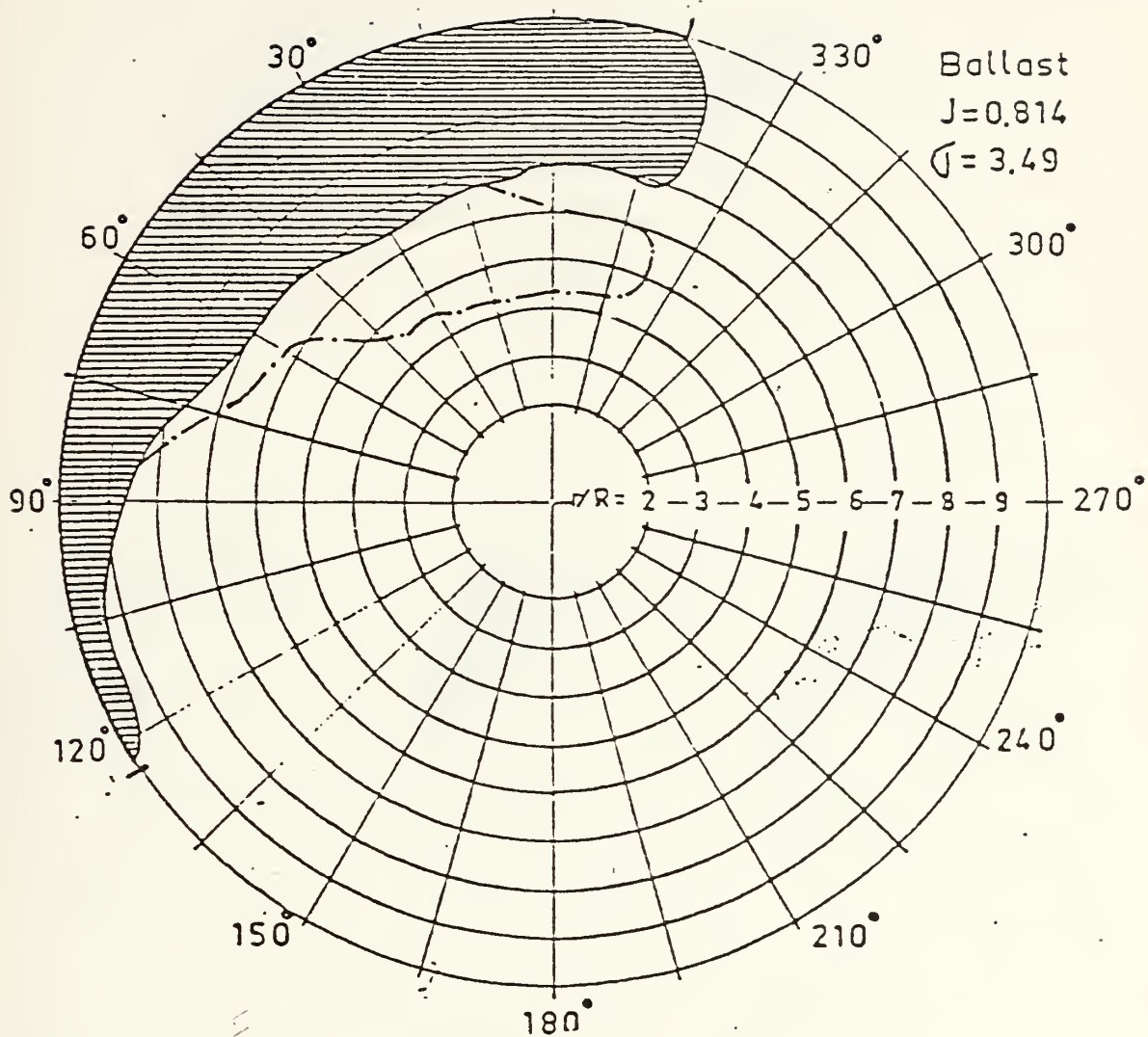
Cavity Volume, Nominal and Effective Wakes

PUF-3 CAVITY VOLUME OUTPUT



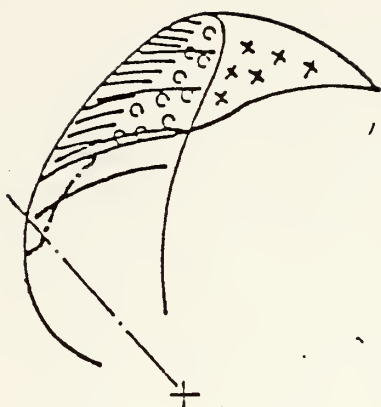
** APPENDIX VII **

SSPA Experimental Propeller Cavitation Extent Plots

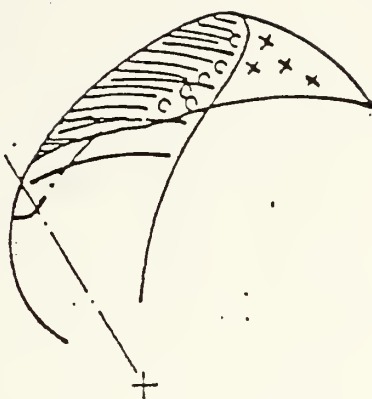


Reference (10)

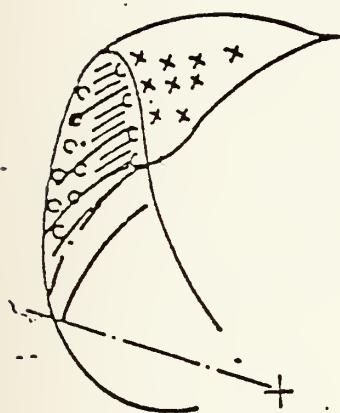
40°



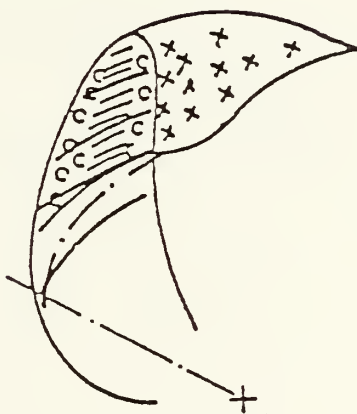
30°



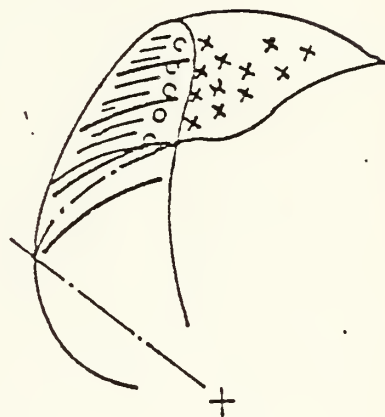
70°



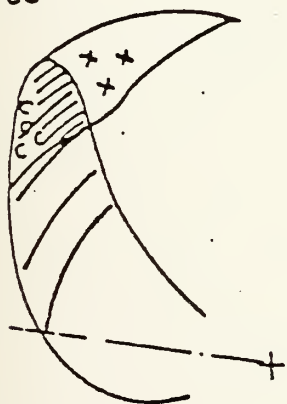
60°



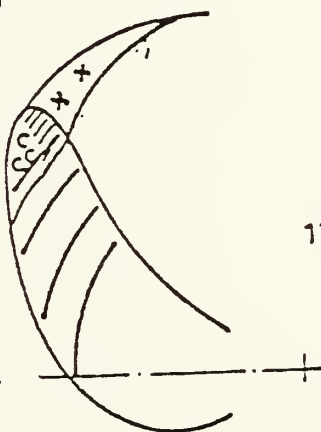
50°



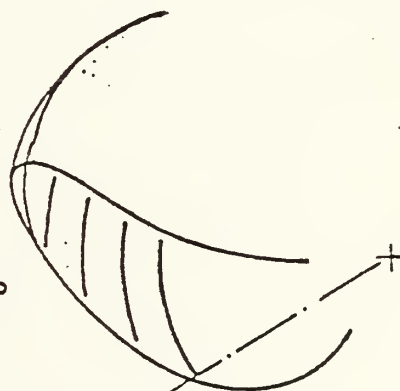
80°



90°



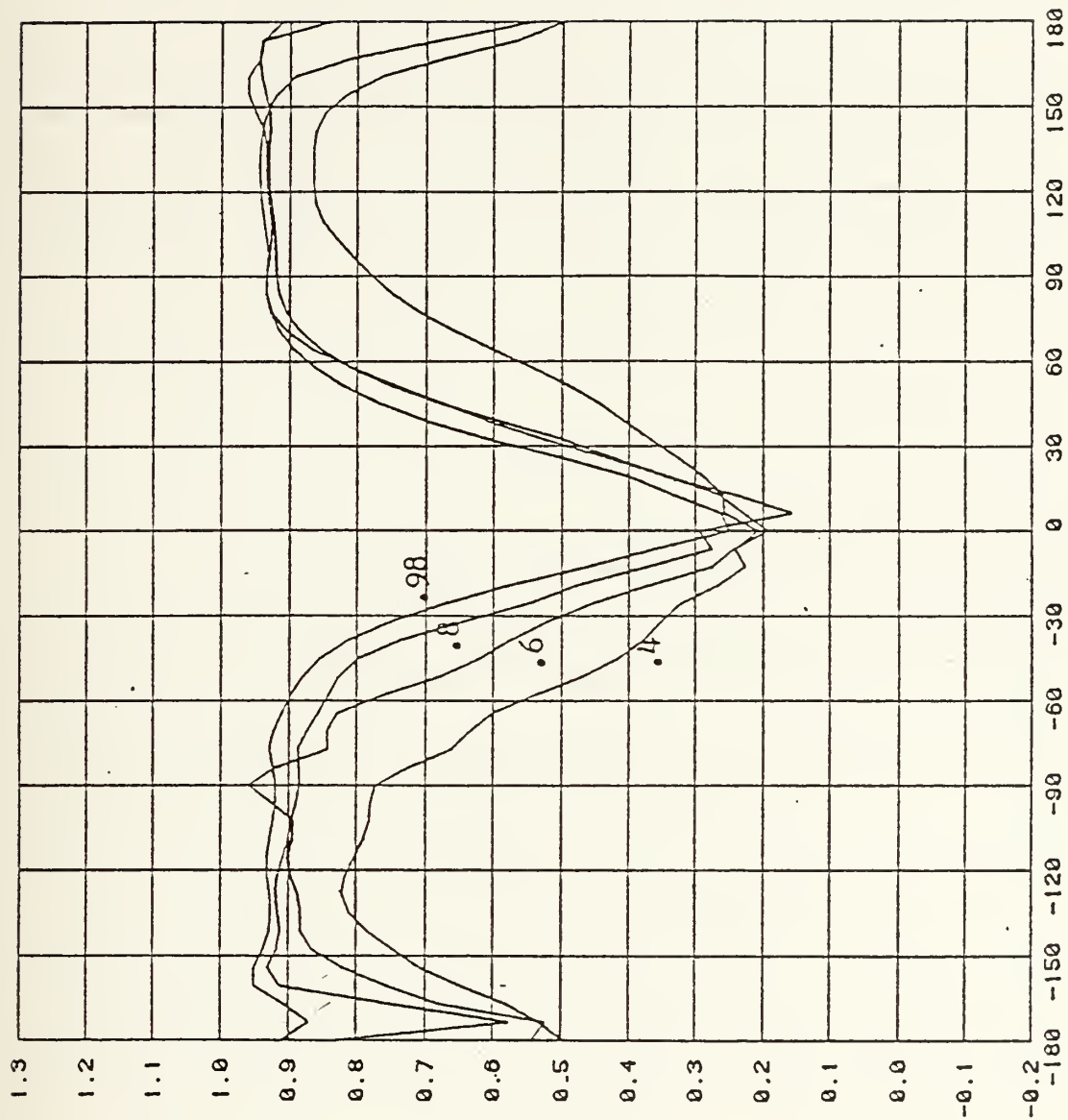
120°



Reference (10)

** APPENDIX VIII **

Nominal Wake Diagram

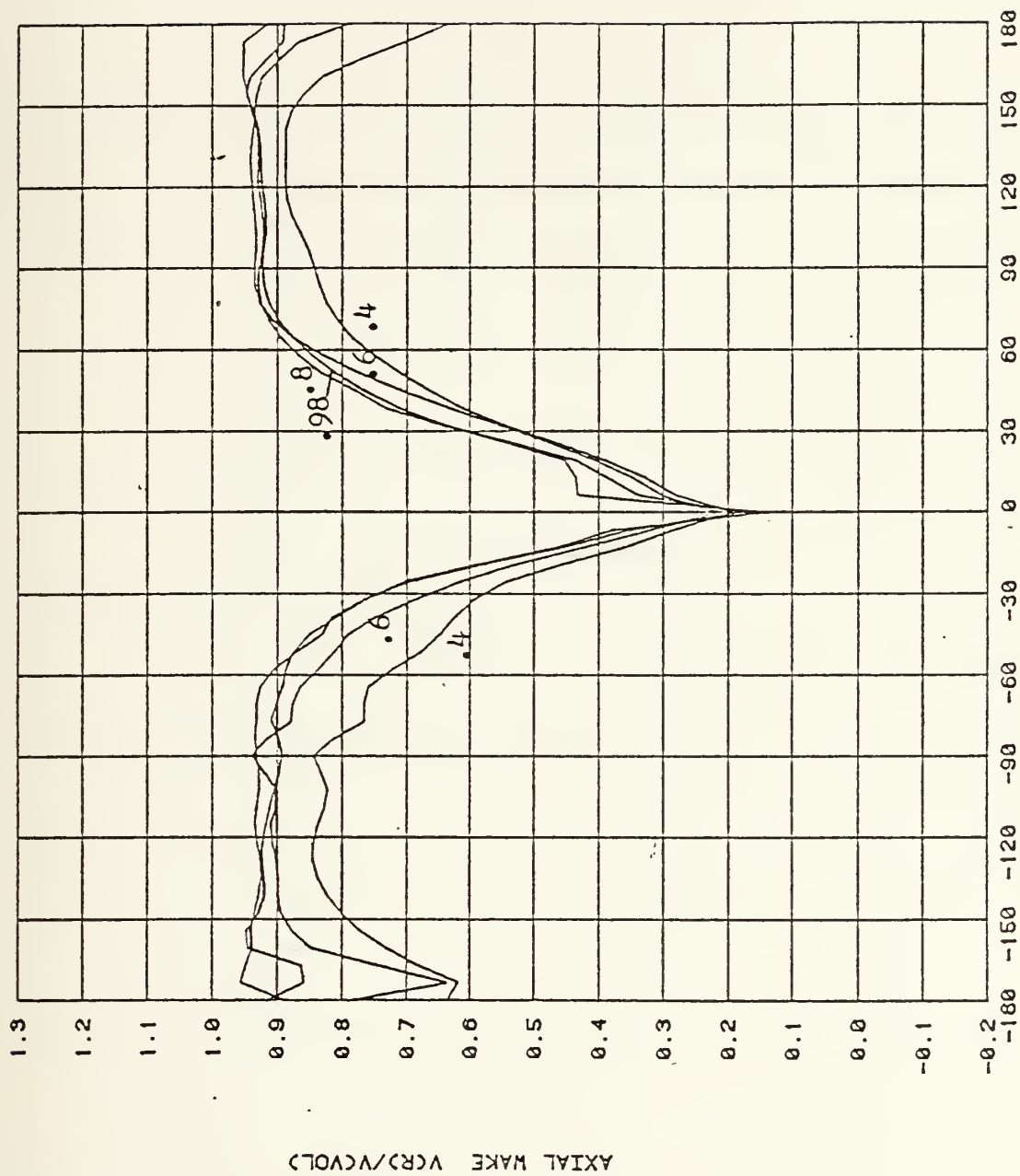


Actual Velocities (I.e., V(VOL)=1.0)

NOMINAL WAKE, .4, .6, .8, .98 RADII

** APPENDIX IX

Effective Wake Diagram



** APPENDIX X **

UFPV Program Listing

C	*****	UFPO0010
C	*	UFPO0020
C	* MAIN PROGRAM IN FPUV, A PROGRAM TO CALCULATE	UFPO0030
C	* UNSTEADY VELOCITY HARMONICS AT FIELD POINTS	UFPO0040
C	* SPECIFIED INTERACTIVELY IN ROTERM.	UFPO0050
C	*	UFPO0060
C	*****	UFPO0070
C		UFPO0080
C		UFPO0090
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO0100
C		UFPO0110
	COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),	UFPO0120
	& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),	UFPO0130
	& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH	UFPO0140
C		UFPO0150
C	*** COMMON TO CONTAIN SINGULARITY STRENGTHS	UFPO0160
C		UFPO0170
	COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV	UFPO0180
	& ,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ	UFPO0190
C		UFPO0200
C	*** COMMON TO CONTAIN FIELD POINT DATA, NBLAOE, IOENT, NSR, AND VSR	UFPO0210
C		UFPO0220
	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO0230
	& OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPLLOT,OIAM	UFPO0240
C		UFPO0250
C	*** COMMON TO CONTAIN VELOCITY DATA	UFPO0260
C		UFPO0270
	COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP	UFPO0280
C		UFPO0290
C	*** COMMON TO CONTAIN HARMONICS COEFFICIENTS	UFPO0300
C		UFPO0310
	COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH	UFPO0320
C		UFPO0330
C	*** ISTOP IS A FLAG SET INTERACTIVELY IN ROTERM TO INDICATE END OF PROUFPO0340	UFPO0350
C		UFPO0360
	ISTOP=0	UFPO0370
C		UFPO0380
C	*** ROFILE WILL READ DISK DATA FILE CREATED BY PUF-2. NOTE THAT NSR	UFPO0390
	PUF-2 INPUT MUST BE DIVISIBLE BY THE NUMBER OF BLAOES, NBLAOE.	UFPO0400
C		UFPO0410
	CALL ROFILE	UFPO0420
C		UFPO0430
C	*** INTERACTIVELY READ PARAMETERS FOR TERMINAL SESSION	UFPO0440
C		UFPO0450
905	FORMAT('/' ENTER NBLAOE, NTORL, IPLLOT, '/'	UFPO0460
	& ' FOR HELP, ENTER ZERO FOR NBLAOE.')	UFPO0470
30	WRITE(6,905)	UFPO0480
	READ(5,*) NBLAOE,NTORL,IPLLOT	UFPO0490
	IF (NBLAOE.LT.0) GOTO 20	UFPO0500
	IF (NBLAOE.EQ.0) GOTO 90	UFPO0510
	IF (IPLLOT.EQ.2) CALL PIEOAT(ISTOP)	UFPO0520
	IF (ISTOP.EQ.1) GOTO 20	UFPO0530
C		

C *** ROTERM GETS DATA INTERACTIVELY FROM TERMINAL AND COMPUTES	UFPO0540
C NECESSARY FIELD POINT GEOMETRY.	UFPO0550
C	UFPO0560
C 10 CALL ROTERM(ISTDP)	UFPO0570
C	UFPO0580
C *** IF OPERATOR INDICATES END OF PROGRAM, ISTOP WILL BE SET TO 1	UFPO0590
C	UFPO0600
C IF (ISTOP.GT.0) GOTO 20	UFPO0610
C	UFPO0620
C *** RUN THROUGH FPSTEP FOR EACH "TIME STEP."	UFPO0630
C	UFPO0640
C DD 100 K=1,NSR	UFPO0650
C	UFPO0660
C *** FPSTEP CALCULATED VELOCITY INDUCED AT FIELD POINT BY ONE BLADE	UFPO0670
C AT "TIME STEP" K	UFPO0680
C	UFPO0690
C 100 CALL FPSTEP(K)	UFPO0700
C	UFPO0710
C *** SUMVEL WILL SUM FIELD POINT VELOCITIES INDUCED BY EACH BLADE.	UFPO0720
C	UFPO0730
C CALL SUMVEL	UFPO0740
C	UFPO0750
C *** FDUVAL WILL GENERATE HARMONIC COEFFICIENTS FOR INDUCED FIELD POINT	UFPO0760
C VELOCITY AS PROPELLER ROTATES.	UFPO0770
C	UFPO0780
C CALL FDUVAL	UFPO0790
C	UFPO0800
C *** PLOTVL PLOTS INDUCED FIELD POINT VELOCITIES AS A FUNCTION OF BLADE	UFPO0810
C ROTATION.	UFPO0820
C	UFPO0830
C 901 FORMAT(' AXIAL COORDINATE',4X,F5.3/' RADIAL COORDINATE',2X,F6.3/	UFPO0840
C & ' ANGULAR COORDINATE',2X,F5.1///' AXIAL HARMONIC',6X,F8.5/	UFPO0850
C & ' RADIAL HARMONIC',5X,F8.5/' TANGENTIAL HARMONIC ',F8.5)	UFPO0860
C WRITE(6,901) XFP,RFP,TZFP,(AMP(1,J),J=1,3)	UFPO0870
C IF (IPLOT.EQ.1) CALL PLOTVL(NBLADE,NSR,NH,TFP,TZFP,U,A,B,AMP,PH,	UFPO0880
C & XFP,RFP,IDENT)	UFPO0890
C GOTO 10	UFPO0900
C 906 FORMAT('/' NBLADE < 0 => STOP'	UFPO0910
C & /8X, '= 0 => HELP'/8X,'> 0 => NUMBER OF BLADES'/	UFPO0920
C & ' NTORL = 0 => TOTAL EFFECT'/8X,'= 1 => THICKNESS ONLY'/	UFPO0930
C & 8X,'= 2 => LOADING ONLY'/ ' IPLOT = 0 => DATA TO TERMINAL'/	UFPO0940
C & 8X,'= 1 => PLDT -- NOT IMPLEMENTED 05/02/82 -- RLJ'	UFPO0950
C & /8X,'= 2 => CREATE PIEWAKE DATA FILE'/)	UFPO0960
C 90 WRITE(6,906)	UFPO0970
C GOTO 30	UFPO0980
C 20 STOP	UFPO0990
C END	UFPO1000

	SUBROUTINE FPVEL(K,XF,YF,ZF,VIX,VIY,VIZ)	UFPO1010
C		UFPO1020
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO1030
C		UFPO1040
	COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),	UFPO1050
	& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),	UFPO1060
	& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH	UFPO1070
C		UFPO1080
C	*** COMMON TO CONTAIN SINGULARITY STRENGTHS	UFPO1090
C		UFPO1100
	COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV	UFPO1110
	& ,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ	UFPO1120
C		UFPO1130
C	*** COMMON TO CONTAIN FIELO POINT DATA, NBLAOE, IOENT, NSR, AND VSR	UFPO1140
C		UFPO1150
	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO1160
	& OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPL0T,OIAM	UFPO1170
C		UFPO1180
	DIMENSION UTX(20,2),UTY(20,2),UTZ(20,2),UBX(20),UBY(20),UBZ(20)	UFPO1190
	& ,VOFWX(20,9),VOFWY(20,9),VOFWZ(20,9)	UFPO1200
C		UFPO1210
C	*** INITIALIZE VARIABLES	UFPO1220
C		UFPO1230
	MIO=MN/2	UFPO1240
	NVFW=NW	UFPO1250
	JMPFW=1	UFPO1260
	WAKX=0.	UFPO1270
	WAKY=0.	UFPO1280
	WAKZ=0.	UFPO1290
	VIX=0.0	UFPO1300
	VIY=0.0	UFPO1310
	VIZ=0.0	UFPO1320
	DO 1 M=1,MM	UFPO1330
	OO 1 N=1,NN	UFPO1340
C	-----CHOROWISE VORTICES ON BLAOE-----	UFPO1350
	IF(NTORL.EQ.1) GO TO 8	UFPO1360
	CALL VORSEG(XF,YF,ZF,X(N,M),Y(N,M),Z(N,M),X(N+1,M),Y(N+1,M),	UFPO1370
	1 Z(N+1,M),CVX,CVY,CVZ,SX,SY,SZ,0)	UFPO1380
C	-----SPANWISE VORTICES AND SOURCES ON BLAOE-----	UFPO1390
8	CALL VORSEG(XF,YF,ZF,X(N,M),Y(N,M),Z(N,M),X(N,M+1),Y(N,M+1),	UFPO1400
	1 Z(N,M+1),SVX,SVY,SVZ,SX,SY,SZ,1)	UFPO1410
	SBNM=SB(N,M)	UFPO1420
	GBNM=GB(N,M,K)	UFPO1430
	GTNM=GT(N,M,K)	UFPO1440
	IF(NTORL.EQ.0) GO TO 9	UFPO1450
	IF(NTORL.EQ.1) GBNM=0.0	UFPO1460
	IF(NTORL.EQ.1) GTNM=0.0	UFPO1470
	IF(NTORL.EQ.2) SBNM=0.0	UFPO1480
9	VIX=VIX+GBNM*SVX+SBNM*SX+GTNM*CVX	UFPO1490
	VIY=VIY+GBNM*SVY+SBNM*SY+GTNM*CVY	UFPO1500
	VIZ=VIZ+GBNM*SVZ+SBNM*SZ+GTNM*CVZ	UFPO1510
1	CONTINUE	UFPO1520
C	-----SEPARATED TIP VORTEX ON BLAOE-----	UFPO1530

IF(NTORL.EQ.1) GO TO 10	UFPO1540
00 2 N=1,NN	UFPO1550
CVX=0.0	UFPO1560
CVY=0.0	UFPO1570
CVZ=0.0	UFPO1580
NTEM=NN-N+1	UFPO1590
00 3 LL=1,NTEM	UFPO1600
L1=LL+1	UFPO1610
CALL VORSEG(XF,YF,ZF,XTIP(N,LL),YTIP(N,LL),ZTIP(N,LL)	UFPO1620
*,XTIP(N,L1),YTIP(N,L1),ZTIP(N,L1),VX,VY,VZ,SX,SY,SZ,O)	UFPO1630
CVX=CVX+VX	UFPO1640
CVY=CVY+VY	UFPO1650
3 CVZ=CVZ+VZ	UFPO1660
VIX=VIX+GB(N,MM,K)*CVX	UFPO1670
VIY=VIY+GB(N,MM,K)*CVY	UFPO1680
2 VIZ=VIZ+GB(N,MM,K)*CVZ	UFPO1690
C-----TRANSITION WAKE-----	UFPO1700
WAKX=0.	UFPO1710
WAKY=0.	UFPO1720
WAKZ=0.	UFPO1730
00 35 N=1,100	UFPO1740
CALL VORSEG(XF,YF,ZF,XH(N,1),YH(N,1),ZH(N,1),XH(N+1,1),YH(N+1,1),	UFPO1750
1 ZH(N+1,1),VX,VY,VZ,SX,SY,SZ,O)	UFPO1760
WAKX=WAKX+VX	UFPO1770
WAKY=WAKY+VY	UFPO1780
35 WAKZ=WAKZ+VZ	UFPO1790
CALL VORSEG(XF,YF,ZF,XH(1,1),O.,O.,XH(101,1),O.,O.,	UFPO1800
& HBWX,HBWY,HBWZ,SX,SY,SZ,O)	UFPO1810
HBWX=-HBWX	UFPO1820
HBWY=-HBWY	UFPO1830
HBWZ=-HBWZ	UFPO1840
11 00 71 M=1,MM	UFPO1850
IF(M.EQ.1) GO TO 72	UFPO1860
00 73 N=1,NW	UFPO1870
UTX(N,1)=UTX(N,2)	UFPO1880
UTY(N,1)=UTY(N,2)	UFPO1890
73 UTZ(N,1)=UTZ(N,2)	UFPO1900
72 00 74 N=1,NW	UFPO1910
CALL VORSEG(XF,YF,ZF,XW(N,M),YW(N,M),ZW(N,M),XW(N,M+1),YW(N,M+1),	UFPO1920
1 ZW(N,M+1),UBX(N),UBY(N),UBZ(N),SX,SY,SZ,O)	UFPO1930
IF(M.GT.1) GO TO 75	UFPO1940
CALL VORSEG(XF,YF,ZF,XW(N,M),YW(N,M),ZW(N,M),XW(N+1,M),YW(N+1,M),	UFPO1950
1 ZW(N+1,M),UTX(N,1),UTY(N,1),UTZ(N,1),SX,SY,SZ,O)	UFPO1960
75 CALL VORSEG(XF,YF,ZF,XW(N,M+1),YW(N,M+1),ZW(N,M+1),XW(N+1,M+1),	UFPO1970
1 YW(N+1,M+1),ZW(N+1,M+1),UTX(N,2),UTY(N,2),UTZ(N,2),SX,SY,SZ,O)	UFPO1980
74 CONTINUE	UFPO1990
00 76 N=1,NW	UFPO2000
C	UFPO2010
C *** COMPUTE INOUCED VELOCITY FROM ONE FORCESHOE AT THIS RADIUS	UFPO2020
C	UFPO2030
BUG=UTX(N,2)-UTX(N,1)+UBX(N)	UFPO2040
CAT=UTY(N,2)-UTY(N,1)+UBY(N)	UFPO2050
OOG=UTZ(N,2)-UTZ(N,1)+UBZ(N)	UFPO2060

	IF(N.NE.NW) GO TO 77	UFPO2070
	IF(M.NE.MID) GO TO 78	UFPO2080
C		UFPO2090
C	*** ALL TRANSITION WAKE VORTICES DISAPPEAR EXCEPT MID, WHICH TURNS	UFPO2100
C	INTO ULTIMATE WAKE VORTICES (TIP AND HUB) BY ASSUMING MEAN LOADING	UFPO2110
C		UFPO2120
	BUG=BUG+WAKX+HBWX	UFPO2130
	CAT=CAT+WAKY+HBWY	UFPO2140
	DOG=DOG+WAKZ+HBWZ	UFPO2150
	GO TO 78	UFPO2160
77	BUG=BUG-UBX(N+1)	UFPO2170
	CAT=CAT-UBY(N+1)	UFPO2180
	DOG=DOG-UBZ(N+1)	UFPO2190
78	VOFWX(N,M)=BUG	UFPO2200
	VOFWY(N,M)=CAT	UFPO2210
	VOFWZ(N,M)=DOG	UFPO2220
76	CONTINUE	UFPO2230
92	DO 93 LN=1,NVFW	UFPO2240
	KKM=K-LN	UFPO2250
	IF (KKM.LE.O) KKM=KKM+NSR	UFPO2260
	VIX=VIX+VOFWX(LN,M)*(GMEAN(M)	UFPO2270
1	+DKFW(LN)=(GAM(M,KKM)-GMEAN(M)))	UFPO2280
	VIIY=VIY+VOFWY(LN,M)*(GMEAN(M)	UFPO2290
1	+DKFW(LN)=(GAM(M,KKM)-GMEAN(M)))	UFPO2300
93	VIZ=VIZ+VOFWZ(LN,M)*(GMEAN(M)	UFPO2310
1	+DKFW(LN)=(GAM(M,KKM)-GMEAN(M)))	UFPO2320
71	CONTINUE	UFPO2330
10	RETURN	UFPO2340
	END	UFPO2350

C	SUBROUTINE RDFILE	UFPO2360
C	*****	UFPO2370
C	*	UFPO2380
C	* SUBROUTINE RDFILE READS DISK FILE14 DATA CREATED *	UFPO2390
C	* BY PUF-2 AND XAVES DATA NEEDED BY FPUV IN *	UFPO2400
C	* COMMONS GEDCOM AND SINCOM. NOTE THAT NSR IN *	UFPO2410
C	* PUF-2 MUST BE DIVISIBLE BY THE NUMBER OF BLADES, *	UFPO2420
C	* NBLADE.	UFPO2430
C	*	UFPO2440
C	*	UFPO2450
C	* NN =>	UFPO2460
C	* MM =>	UFPO2470
C	* NW => NUMBER OF VORTICES IN KEY WAKE	UFPO2480
C	* NBLADE => NUMBER OF PROPELLER BLADES	UFPO2490
C	* NVC => NO OF CHORD VORTICES, OTHER BLADES	UFPO2500
C	* NVS => NO OF SPAN VORTICES, OTHER BLADES	UFPO2510
C	* NVW => NO OF WAKE VORTICES, OTHER WAKE	UFPO2520
C	* NSR => NUMBER OF TIME STEPS/REVOLUTION	UFPO2530
C	* NWK => NUMBER OF RADII IN WAKE HARMONICS	UFPO2540
C	*	UFPO2550
C	*****	UFPO2560
C		UFPO2570
C		UFPO2580
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO2590
C	COMMON / GEOCOM / MM,MN,NN,NW,NTDRL,X(12,10),Y(12,10),Z(12,10),	UFPO2610
C	& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),	UFPO2620
C	& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH	UFPO2630
C		UFPO2640
C	*** COMMON TO CONTAIN SINGULARITY STRENGTHS	UFPO2650
C	COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV	UFPO2670
C	& ,GAM(9,60),GMEAN(9),DKFW(20),HBWX,HBWY,HBWZ	UFPO2680
C		UFPO2690
C	*** COMMON TO CONTAIN FIELD POINT DATA, NBLADE, IDENT, NSR, AND VSR	UFPO2700
C	COMMON / FPCOM / NBLADE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO2720
C	& DBLADE,DTFP,IDENT(18),TFP(60),VSR,IPL0T,DIAM	UFPO2730
C		UFPO2740
C		UFPO2750
C	DIMENSION RZ(10),R(9),CHDCP(9),CHORZ(10),PHIB(90)	UFPO2760
C	DIMENSION XD(6,4,6),YO(6,4,6),ZD(6,4,6)	UFPO2770
C	DIMENSION RWK(11),AWA(16,11),BWA(16,11),AWR(16,11),BWR(16,11),	UFPO2780
C	& AWT(16,11),BWT(16,11)	UFPO2790
C	DIMENSION XWO(5,4,6),YWO(5,4,6),ZWO(5,4,6)	UFPO2800
C	DIMENSION SBO(5,3),GBC(5,3,60),GAMWC(4,3,60),GTC(5,4,60),	UFPO2810
C	& GAMC(3,60),GMEANO(3),DKOW(4)	UFPO2820
C	READ(14) AJ	UFPO2830
C	IF (AJ.GE.2.0) STDP	UFPO2840
C	READ(14) ISTDY,NBLADE,MM,NN,NW,NVC,NVS,NVW,NSR,NWK	UFPO2850
C		UFPO2860
C	*** SET CONSTANT FOR GRID	UFPO2870
C		UFPO2880

NM=NN-1	UFPO2890
MN=MM+1	UFPO2900
NP=NN+1	UFPO2910
NZ=NW+1	UFPO2920
NVCP=NVC+1	UFPO2930
NVSP=NVS+1	UFPO2940
NBLO=NBLADE-1	UFPO2950
C	UFPO2960
C *** SET CONSTANTS FOR UNSTEADY PROBLEM	UFPO2970
C	UFPO2980
IF (ISTDY.EQ.1) NSR=1	UFPO2990
NVWP=N VW+1	UFPO3000
NTB=NN-MM	UFPO3010
C	UFPO3020
C *** READ KEY BLADE GEOMETRY	UFPO3030
C	UFPO3040
READ(14) ((X(N,M),N=1,NP),M=1,MN),	UFPO3050
2 ((Y(N,M),N=1,NP),M=1,MN),	UFPO3060
3 ((Z(N,M),N=1,NP),M=1,MN),	UFPO3070
4 (RZ(M),M=1,MN),	UFPO3080
5 (R(M),M=1,MM),	UFPO3090
6 (CHOC(M),M=1,MM),	UFPO3100
7 (CHORZ(M),M=1,MN),	UFPO3110
8 (PHIB(K),K=1,NTB)	UFPO3120
C	UFPO3130
C *** READ BLADE GEOMETRY OF OTHER BLADES	UFPO3140
C	UFPO3150
IF (NBLADE.LE.1) GOTO 11	UFPO3160
READ(14) (((XD(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLD),	UFPO3170
2 (((YO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),	UFPO3180
3 (((ZO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLD)	UFPO3190
11 CONTINUE	UFPO3200
C	UFPO3210
C *** READ WAKE RADII AND WAKE HARMONICS	UFPO3220
C	UFPO3230
READ(14) (RWK(M),M=1,NWK)	UFPO3240
READ(14) ((AWA(I,M),I=1,16),M=1,NWK),	UFPO3250
2 ((BWA(I,M),I=1,16),M=1,NWK),	UFPO3260
3 ((AWR(I,M),I=1,16),M=1,NWK),	UFPO3270
4 ((BWR(I,M),I=1,16),M=1,NWK),	UFPO3280
5 ((AWT(I,M),I=1,16),M=1,NWK),	UFPO3290
6 ((BWT(I,M),I=1,16),M=1,NWK)	UFPO3300
C	UFPO3310
C *** READ VARIOUS PROPELLER/WAKE CONSTANTS	UFPO3320
C	UFPO3330
READ(14) WAKE,RPM,DIAM,RH,CDRAG,VSR,UR,(IDENT(N),N=1,18)	UFPO3340
C	UFPO3350
C *** READ KEY WAKE GEOMETRY	UFPO3360
C	UFPO3370
READ(14) ((XW(N,M),N=1,NZ),M=1,MN),	UFPO3380
2 ((YW(N,M),N=1,NZ),M=1,MN),	UFPO3390
3 ((ZW(N,M),N=1,NZ),M=1,MN),	UFPO3400
4 ((XH(N,K),N=1,101),K=1,NBLADE),	UFPO3410

5	((YH(N,K),N=1,101),K=1,NBLADE),	UFPO3420
6	((ZH(N,K),N=1,101),K=1,NBLADE),	UFPO3430
7	((XTIP(N,L),L=1,NP),N=1,NN),	UFPO3440
8	((YTIP(N,L),L=1,NP),N=1,NN),	UFPO3450
9	((ZTIP(N,L),L=1,NP),N=1,NN),	UFPO3460
C		UFPO3470
C	*** READ KEY BLADE SINGULARITY STRENGTHS	UFPO3480
C		UFPO3490
*	((SB(N,M),N=1,NN),M=1,MM),	UFPO3500
1	((GB(N,M,KK),N=1,NN),M=1,MM),KK=1,NSR),	UFPO3510
2	((GT(N,M,KK),N=1,NN),M=1,MN),KK=1,NSR),	UFPO3520
3	((GAM(M,KK),M=1,MM),KK=1,NSR)	UFPO3530
C		UFPO3540
C	*** OTHER WAKE GEOMETRY	UFPO3550
C		UFPO3560
	IF (NBLADE.LE.1) GOTO 12	UFPO3570
	READ(14) (((XWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),	UFPO3580
2	((YWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),	UFPO3590
3	((ZWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),	UFPO3600
C		UFPO3610
C	*** OTHER BLADE SINGULARITIES	UFPO3620
C		UFPO3630
4	((SBO(N,M),N=1,NVC),M=1,NVS),	UFPO3640
5	((GBC(N,M,KK),N=1,NVC),M=1,NVS),KK=1,NSR),	UFPO3650
6	((GAMWC(N,M,KK),N=1,NVW),M=1,NVS),KK=1,NSR),	UFPO3660
7	((GTC(N,M,KK),N=1,NVC),M=1,NVSP),KK=1,NSR),	UFPO3670
8	((GAMC(M,KK),M=1,NVS),KK=1,NSR),	UFPO3680
9	(GMEAN(M),M=1,MM),	UFPO3690
*	(GMEANO(M),M=1,NVS),	UFPO3700
1	(DKFW(N),N=1,NW),	UFPO3710
2	(DKOW(N),N=1,NVW)	UFPO3720
12	CONTINUE	UFPO3730
99	GOTO 100	UFPO3740
C		UFPO3750
C	*** DIAGNOSTICS ***	UFPO3760
C		UFPO3770
901	FORMAT(16I5)	UFPO3780
902	FORMAT(8F10.5)	UFPO3790
903	FORMAT(18A4)	UFPO3800
	WRITE(1,902) AJ	UFPO3810
	WRITE(1,901) ISTDY,NBLADE,MM,NN,NW,NVC,NVS,NVW,NSR,NWK	UFPO3820
	WRITE(1,902) ((X(N,M),N=1,NP),M=1,MN),	UFPO3830
2	((Y(N,M),N=1,NP),M=1,MN),	UFPO3840
3	((Z(N,M),N=1,NP),M=1,MN),	UFPO3850
4	(RZ(M),M=1,MN),	UFPO3860
5	(R(M),M=1,MM),	UFPO3870
6	(CHOC(M),M=1,MM),	UFPO3880
7	(CHORZ(M),M=1,MN),	UFPO3890
8	(PHIB(K),K=1,NTB)	UFPO3900
	IF (NBLADE.LE.1) GOTO 21	UFPO3910
	WRITE(1,902) (((XO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),	UFPO3920
2	((YO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),	UFPO3930
3	((ZO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO)	UFPO3940

21	CONTINUE	UFPO3950
	WRITE(1,902) (RWK(M),M=1,NWK)	UFPO3960
	WRITE(1,902) ((AWA(I,M),I=1,16),M=1,NWK),	UFPO3970
3	((BWA(I,M),I=1,16),M=1,NWK),	UFPO3980
4	((AWR(I,M),I=1,16),M=1,NWK),	UFPO3990
5	((BWR(I,M),I=1,16),M=1,NWK),	UFPO4000
6	((AWT(I,M),I=1,16),M=1,NWK),	UFPO4010
7	((BWT(I,M),I=1,16),M=1,NWK)	UFPO4020
	WRITE(1,902) WAKE,RPM,DIAM,RH,CDRAG,VSR,UR	UFPO4030
	WRITE(1,903) (IDENT(N),N=1,18)	UFPO4040
	WRITE(1,902) ((XW(N,M),N=1,NZ),M=1,MN),	UFPO4050
2	((YW(N,M),N=1,NZ),M=1,MN),	UFPO4060
3	((ZW(N,M),N=1,NZ),M=1,MN),	UFPO4070
4	((XH(N,K),N=1,101),K=1,NBLADE),	UFPO4080
5	((YH(N,K),N=1,101),K=1,NBLADE),	UFPO4090
6	((ZH(N,K),N=1,101),K=1,NBLADE),	UFPO4100
7	((XTIP(N,L),L=1,NP),N=1,NN),	UFPO4110
8	((YTIP(N,L),L=1,NP),N=1,NN),	UFPO4120
9	((ZTIP(N,L),L=1,NP),N=1,NN)	UFPO4130
	WRITE(1,902) ((SB(N,M),N=1,NN),M=1,MM),	UFPO4140
2	((GB(N,M,KK),N=1,NN),M=1,MM),KK=1,NSR),	UFPO4150
3	((GT(N,M,KK),N=1,NN),M=1,MM),KK=1,NSR),	UFPO4160
4	((GAM(M,KK),M=1,MM),KK=1,NSR)	UFPO4170
	IF (NBLADE.LE.1) GOTO 22	UFPO4180
	WRITE(1,902) (((XWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),	UFPO4190
2	((YWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),	UFPO4200
3	((ZWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO)	UFPO4210
	WRITE(1,902) ((SBD(N,M),N=1,NVC),M=1,NVS),	UFPO4220
2	((GBC(N,M,KK),N=1,NVC),M=1,NVS),KK=1,NSR),	UFPO4230
3	((GAMWC(N,M,KK),N=1,NVW),M=1,NVS),KK=1,NSR),	UFPO4240
4	((GTC(N,M,KK),N=1,NVC),M=1,NVSP),KK=1,NSR),	UFPO4250
5	((GAMC(M,KK),M=1,NVS),KK=1,NSR),	UFPO4260
6	(GMEAN(M),M=1,MM),	UFPO4270
7	(GMEANO(M),M=1,NVS),	UFPO4280
8	(DKFW(N),N=1,NW),	UFPO4290
9	(DKOW(N),N=1,NVW)	UFPO4300
22	CONTINUE	UFPO4310
100	RETURN	UFPO4320
	END	UFPO4330

C	SUBROUTINE ROTERM(ISTOP)	UFPO4340
C	*****	UFPO4350
C	*	UFPO4360
C	* SUBROUTINE ROTERM READS IN BLAOE AND GEOMETRY *	UFPO4370
C	* DATA INTERACTIVELY AND COMPUTES NECESSARY FIELD *	UFPO4380
C	* POINT GEOMETRY. *	UFPO4390
C	*	UFPO4400
C	* NBLAOE => NUMBER OF BLAOES *	UFPO4410
C	* NTORL = 0 => CONSIDER THICKNESS AND LOADING *	UFPO4420
C	* = 1 => CONSIDER THICKNESS ONLY *	UFPO4430
C	* = 2 => CONSIDER LOADING ONLY *	UFPO4440
C	* XFP => AXIAL FIELD POINT COORDINATE, *	UFPO4450
C	* POSITIVE DOWNSTREAM *	UFPO4460
C	* RFP => RADIAL FIELD POINT COORDINATE *	UFPO4470
C	* TZFP => INITIAL FIELD POINT ANGLE IN DEG. *	UFPO4480
C	* YBAR,ZBAR => Y AND Z FIELD POINT COORDINATES *	UFPO4490
C	* TH => ANGLE FROM TIME STEP INPUT *	UFPO4500
C	* YB,ZB => Y,Z FIELD PT COORD. BEFORE TIME ST *	UFPO4510
C	*	UFPO4520
C	*****	UFPO4530
C		UFPO4540
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO4550
C		UFPO4560
	COMMON / GEODIM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),	UFPO4570
	& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),	UFPO4580
	& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH	UFPO4590
C		UFPO4600
C	*** COMMON TO CONTAIN FIELD POINT DATA, NBLAOE, IDENT, NSR, AND VSR	UFPO4610
C		UFPO4620
	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO4630
	& OBLAOE,OTFP,IDENT(18),TFP(60),VSR,IPL0T,OIAM	UFPO4640
C		UFPO4650
C		UFPO4660
901	FORMAT('/' ENTER XFP, RFP, TZFP.')	UFPO4670
	& ' FOR HELP, ENTER -11. FOR XFP.')	UFPO4680
10	WRITE(6,901)	UFPO4690
	READ(5,*) XFP,RFP,TZFP	UFPO4700
	IF (XFP.LT.-10.) GOTO 90	UFPO4710
	IF (TZFP.LT.0.0) GOTO 70	UFPO4720
	IF (RFP.GT.0.) GOTO 60	UFPO4730
20	WRITE(6,902)	UFPO4740
902	FORMAT('/' ENTER NBW,N,M,KK.')	UFPO4750
	READ(5,*) NBW,N,M,KK	UFPO4760
	TH=6.283185*FLOAT(KK-1)/FLOAT(NSR)	UFPO4770
	IF (NBW.EQ.0) GOTO 80	UFPO4780
C		UFPO4790
C	*** FIELD POINT CENTERED ON BLAOE GRID.	UFPO4800
C		UFPO4810
	IF(NBW.EQ.1) CALL FPGRID(X,Y,Z,N,M,KK,12)	UFPO4820
C		UFPO4830
C	*** FIELD POINT CENTERED ON TRANSITION WAKE GRID.	UFPO4840
C		UFPO4850
	IF(NBW.EQ.2) CALL FPGRID(XW,YW,ZW,N,M,KK,21)	UFPO4860

IF(NBW.NE.1.AND.NBW.NE.2) GOTO 10	UFPO4870
60 DTFP=360.0/FLOAT(NSR)	UFPO4880
RETURN	UFPO4890
70 ISTOP=1	UFPO4900
RETURN	UFPO4910
80 WRITE(6,903)	UFPO4920
903 FORMAT(/' NBW = 0 => HELP'/5X,' = 1 => BLADE WAKE GRID'/5X,	UFPO4930
& ' = 2 => TRANSITION WAKE GRID'/' N',7X,'=> CHOROWISE COORDINATE'/'	UFPO4940
& ' M',7X,'=> SPANWISE COORDINATE'/' KK',6X,	UFPO4950
& '=> TIME STEPS ROTATED + 1'/)	UFPO4960
GOTO 20	UFPO4970
90 WRITE(6,904)	UFPO4980
904 FORMAT(/' XFP <-10=> HELP'/8X,	UFPO4990
& '>-10=> AXIAL FIELD POINT COORDINATE, POSITIVE DOWNSTREAM'/'	UFPO5000
& ' RFP < 0 => SELECT COORDINATE USING BLADE OR WAKE GRID'/'	UFPO5010
& 8X,'> 0 => RADIAL FIELD POINT COORDINATE'/'	UFPO5020
& ' TZFP < 0 => STOP'/8X,'> 0 => ANGULAR FIELD POINT COORDINATE'/'	UFPO5030
& /)	UFPO5040
GOTO 10	UFPO5050
END	UFPO5060

C	SUBROUTINE FPGRIO(XX,YY,ZZ,N,M,KK,NDIM)	UFPO5070
C	*****	UFPO5080
C	*	UFPO5090
C	* SUBROUTINE GRID CALCULATES X,R, AND THETA FIELD *	UFPO5100
C	* PDINT GEDMETRY CDMONENTS GIVEN GRIO VECTORS *	UFPO5110
C	* X,Y, AND Z FOR BLAOE OR WAKE, GRIO CODROINATES *	UFPO5120
C	* N AND M, AND TIME STEP KK. *	UFPO5130
C	*	UFPO5140
C	* XX,YY,ZZ => GRIO GEDMETRY VECTDRS *	UFPO5150
C	* N,M => GRIO INDICES SELECTED FOR VELOCITY *	UFPO5160
C	* KK => TIME STEP. NUMBER OF ANGLE STEPS + 1 *	UFPO5170
C	* XFP,YBAR,ZBAR,TZFP,RFP *	UFPO5180
C	* => X,Y,Z,THETA,RADIAL FIELD PT CODRO *	UFPO5190
C	* OBLADE => ANGLE BETWEEN BLAOES *	UFPO5200
C	* TH => ANGLE OF TIME STEP *	UFPO5210
C	* NOIM => FIRST DIMENSION OF GRIDS PASSED *	UFPO5220
C	*	UFPO5230
C	*****	UFPO5240
C	*** CDMON TO CONTAIN FIELD PDINT DATA, NBLAOE, IDENT, NSR, AND VSR	UFPO5250
C		UFPO5260
C	CDMON / FPCDM / NBLADE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO5270
C	& OBLADE,DTFP,IDENT(18),TFP(60),VSR,IPLDT,DIAM .	UFPO5280
C		UFPO5290
C	DIMENSION XX(NDIM,10),YY(NDIM,10),ZZ(NOIM,10)	UFPO5300
C		UFPO5310
C	*** COMPUTE X,Y,Z CODROINATES IN ZERO ANGLE PSDITION	UFPO5320
C		UFPO5330
C	TH=6.283185*FLDAT(KK-1)/FLDAT(NSR)	UFPO5340
C	THDEG=TH*360./6.283185	UFPO5350
C	OBLADE=360.O/FLDAT(NBLAOE)	UFPO5360
C	XFP=0.25*(XX(N,M) + XX(N+1,M) + XX(N,M+1) + XX(N+1,M+1))	UFPO5370
C	YB =0.25*(YY(N,M) + YY(N+1,M) + YY(N,M+1) + YY(N+1,M+1))	UFPO5380
C	ZB =0.25*(ZZ(N,M) + ZZ(N+1,M) + ZZ(N,M+1) + ZZ(N+1,M+1))	UFPO5390
C		UFPO5400
C		UFPO5410
C	*** CONVERT TO CYLINDRICAL CODROINATES	UFPO5420
C		UFPO5430
C	RFP=SQRT(YB*YB+ZB*ZB)	UFPO5440
C	TZFP=ATAN2(ZB,YB)/1.7453293E-02	UFPO5450
C	TZFP=TZFP+THDEG	UFPO5460
C		UFPO5470
C	*** BRANCH IF GENERATING PIEWAKE DATA	UFPO5480
C		UFPO5490
C	IF (IPLDT.EQ.2) GOTO 200	UFPO5500
C	TZFP=180.O/FLDAT(NBLAOE) - ATAN2(ZB,YB)/1.7453293E-02	UFPO5510
C		UFPO5520
C	*** KEEP TZFP BETWEEN 0 AND ONE BLAOE ANGLE	UFPO5530
C		UFPO5540
C	OD 100 K=1,NBLAOE	UFPO5550
C	IF(TZFP.LT.O.O) TZFP=TZFP+OBLADE	UFPO5560
C	IF(TZFP.GT.OBLADE) TZFP=TZFP-OBLADE	UFPO5570
100	CONTINUE	UFPO5580
C		UFPO5590

C *** ROTATE BLADE FOR PIEWAKE DATA
C
200 YBAR=YB*COS(TH) - ZB*SIN(TH)
ZBAR=YB*SIN(TH) + ZB*COS(TH)
300 RETURN
END

UFPO5600
UFPO5610
UFPO5620
UFPO5630
UFPO5640
UFPO5650

C	SUBROUTINE FPSTEP(K)	UFPO5660
C	*****	UFPO5670
C	*	UFPO5680
C	* FPSTEP IS A SUBROUTINE TO CALCULATE THE FIEL	UFPO5690
C	* POINT VELOCITIES INOUCEO BY ONE BLADE AT "TIME	UFPO5700
C	* STEP", K. THE "DIRTY WORK" IS OONE IN	UFPO5710
C	* SUBROUTINE FPVEL.	UFPO5720
C	*	UFPO5730
C	* TFP => ANGULAR FIEL POINT COOROINATE, DEG.	UFPO5740
C	* TRAO => ANGULAR FIEL POINT COOROINATE, RAO.	UFPO5750
C	* YFP,ZFP => Y AND Z FIEL POINT COOROINATES	UFPO5760
C	* VIX,VIY,VIZ,VIR,VIT => INOUCEO VELOCITY COMPONENT*	UFPO5770
C	*	UFPO5780
C	*****	UFPO5790
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO5800
C	COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),	UFPO5810
C	& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),	UFPO5820
C	& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH	UFPO5830
C	*** COMMON TO CONTAIN SINGULARITY STRENGTHS	UFPO5840
C	COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV	UFPO5850
C	& ,GAM(9,60),GMEAN(9),OKFW(20),HSWX,HBWY,HBWZ	UFPO5860
C	*** COMMON TO CONTAIN FIEL POINT OATA, NBLAOE, IOENT, NSR, AND VSR	UFPO5870
C	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO5880
C	& OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPL0T,OIAM	UFPO5890
C	*** COMMON TO CONTAIN VELOCITY OATA	UFPO5900
C	COMMON / VELCDM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP	UFPO5910
C	*** CALCULATE CARTESIAN FIEL POINT COOROINATES TO FEEO FPVEL	UFPO5920
C	TFP(K)=FLOAT(K-1)*OTFP-TZFP	UFPO5930
C	TRAD=TFP(K)*1.7453293E-02	UFPO5940
C	COSFP=COS(TRAD)	UFPO5950
C	SINFP=SIN(TRAO)	UFPO5960
C	YFP=RFP*COSFP	UFPO5970
C	ZFP=RFP*SINFP	UFPO5980
C	*** FPVEL WILL CALCULATE VIX, VIY, AND VIZ FOR FIEL POINT	UFPO5990
C	CALL FPVEL(K,XFP,YFP,ZFP,VIX(K),VIY,VIZ)	UFPO6000
C	VIR(K)=VIY*COSFP + VIZ*SINFP	UFPO6010
C	VIT(K)=VIZ*COSFP - VIY*SINFP	UFPO6020
C	RETURN	UFPO6030
C	END	UFPO6040
		UFPO6050
		UFPO6060
		UFPO6070
		UFPO6080
		UFPO6090
		UFPO6100
		UFPO6110
		UFPO6120
		UFPO6130
		UFPO6140
		UFPO6150
		UFPO6160

C	SUBROUTINE SUMVEL	UFPO6170
C	*****	UFPO6180
C	*	UFPO6190
C	* SUBROUTINE SUMVEL SUMS VELOCITIES FROM EACH	UFPO6200
C	* BLADE, CALCULATED IN FPSTEP, TO PRODUCE THE	UFPO6210
C	* TOTAL PROPELLER-INDUCED FIELD POINT VELOCITY.	UFPO6220
C	*	UFPO6230
C	* VIX,VIR,VIT => VELOCITY COMPONENTS INOUCED BY	UFPO6240
C	* ONE BLAOE AT "TIME STEP" K	UFPO6250
C	* U(N,J) => VELOCITY INOUCED BY ALL BLAOES	UFPO6260
C	* AT "TIME STEP" N	UFPO6270
C	* J=0 => AXIAL COMPONENT	UFPO6280
C	* J=1 => RAOIAL COMPONENT	UFPO6290
C	* J=2 => TANGENTIAL COMPONENT	UFPO6300
C	* NSAMP => NUMBER OF "TIME STEPS" IN ONE	UFPO6310
C	* BLAOE ANGLE (= NUMBER OF	UFPO6320
C	* U VELOCITIES COMPUTED.)	UFPO6330
C	*	UFPO6340
C	*****	UFPO6350
C	*** COMMON TO CONTAIN FIELD POINT DATA, NBLAOE, IDENT, NSR, AND VSR	UFPO6360
C	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO6370
C	& OBLAOE,OTFP,IDENT(18),TFP(60),VSR,IPL0T,OIAM	UFPO6380
C		UFPO6390
C	*** COMMON TO CONTAIN VELOCITY DATA	UFPO6400
C	COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP	UFPO6410
C		UFPO6420
C	*** AT ANY GIVEN REAL TIME, EACH BLAOE CAN BE CONSIDERED TO HAVE	UFPO6430
C	TRAVELLED THROUGH A CERTAIN NUMBER OF "TIME STEPS".	UFPO6440
C	BY SUMMING VIX, VIR, AND VIT AT BLAOE RATE, WE CAN INCLUOE	UFPO6450
C	THE VELOCITIES INDUCED BY EACH BLAOE INTO ONE VELOCITY	UFPO6460
C	INOUCED BY THE ENTIRE PROPELLER.	UFPO6470
C		UFPO6480
C	NSAMP=NSR/NBLAOE	UFPO6490
C	DO 20 N=1,NSAMP	UFPO6500
C	DO 10 J=1,3	UFPO6510
C	10 U(N,J)=0.0	UFPO6520
C	DO 20 K=1,NBLAOE	UFPO6530
C	L=N+(K-1)*NSAMP	UFPO6540
C	U(N,1)=U(N,1)+VIX(L)	UFPO6550
C	U(N,2)=U(N,2)+VIR(L)	UFPO6560
C	20 U(N,3)=U(N,3)+VIT(L)	UFPO6570
C		UFPO6580
C	*** NOW CONVERT FROM REFERENCE VELOCITY TO SHIP VELOCITY	UFPO6590
C		UFPO6600
C	DO 30 N=1,NSAMP	UFPO6610
C	DO 30 J=1,3	UFPO6620
C	30 U(N,J)=U(N,J)/VSR	UFPO6630
C	RETURN	UFPO6640
C	END	UFPO6650
		UFPO6660
		UFPO6670
		UFPO6680

C	SUBROUTINE FOUICAL	UFPO6690
C	*****	UFPO6700
C	*	UFPO6710
C	* SUBROUTINE FOUICAL TAKES THE PROPELLER INOUCED	UFPO6720
C	* VELOCITIES CALCULATED IN SUMVEL AND PERFORMS	UFPO6730
C	* A HARMONIC ANALYSIS ON THEM. IT CALCULATES	UFPO6740
C	* FOURIER COEFFICIENTS FOR BOTH COSINE (A) AND	UFPO6750
C	* SINE (B) AND ALSO COSINE AMPLITUOE (AMP) AND	UFPO6760
C	* PHASE ANGLE (PH).	UFPO6770
C	*	UFPO6780
C	* NH => NUMBER OF HARMONICS CALCULATED	UFPO6790
C	* A(N,J),B(N,J) => FOURIER COEFFICIENTS	UFPO6800
C	* A => COSINE COEFFICIENTS	UFPO6810
C	* B => SINE COEFFICIENTS	UFPO6820
C	* N => HARMONIC NUMBER UP TO 15	UFPO6830
C	* J = 1 => AXIAL VELOCITY	UFPO6840
C	* = 2 => RAOIAL VELOCITY	UFPO6850
C	* = 3 => TANGENTIAL VELOCITY	UFPO6860
C	* AMP(N,J) => FOURIER COEFFICIENT AMPLITUDE	UFPO6870
C	* N AND J SUBSCRIPTS AS FOR A, B	UFPO6880
C	* PH(N,J) => PHASE ANGLE ASSOCIATED WITH AMP	UFPO6890
C	* NFB => NUMBER OF FIELO POINTS / BLAOE	UFPO6900
C	*	UFPO6910
C	*****	UFPO6920
C	*** COMMON TO CONTAIN FIELO POINT OATA, NBLAOE, IDENT, NSR, AND VSR	UFPO6930
C	COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,	UFPO6940
C	& OBLAOE,OTFP,IDENT(18),TFP(60),VSR,IPL0T,OIAM	UFPO6950
C	*** COMMON TO CONTAIN VELOCITY DATA	UFPO6960
C	COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP	UFPO6970
C	*** COMMON TO CONTAIN HARMONICS COEFFICIENTS	UFPO6980
C	COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH	UFPO6990
C	*** FIRST CALCULATE A AND B	UFPO7000
C	NFB=NSR/NBLAOE	UFPO7010
C	NHMAX=NFB/2+1	UFPO7020
C	NH=MINO(NHMAX,15)	UFPO7030
C	STEP=6.283185/NFB	UFPO7040
C	ANFB=2.0/NFB	UFPO7050
C	BNFB=1.0/NFB	UFPO7060
C	00 11 K=1,NH	UFPO7070
C	00 10 J=1,3	UFPO7080
C	A(K,J)=0.0	UFPO7090
10	B(K,J)=0.0	UFPO7100
C	00 9 N=1,NFB	UFPO7110
C	T=((N-1)*(K-1))*STEP	UFPO7120
C	ST=SIN(T)	UFPO7130
		UFPO7140
		UFPO7150
		UFPO7160
		UFPO7170
		UFPO7180
		UFPO7190
		UFPO7200
		UFPO7210

CT=COS(T)	UFPO7220
DO 9 J=1,3	UFPO7230
A(K,J)=A(K,J)+U(N,J)*CT	UFPO7240
9 B(K,J)=B(K,J)+U(N,J)*ST	UFPO7250
C	UFPO7260
C *** CONVERT A AND B INTO AMP AND PH	UFPO7270
C	UFPO7280
DO 11 J=1,3	UFPO7290
IF(K.EQ.1) A(1,J)=A(1,J)*BNFB	UFPO7300
IF(K.EQ.NHMAX) GO TO 12	UFPO7310
IF(K.GT.1) A(K,J)=A(K,J)*ANFB	UFPO7320
IF(K.GT.1) B(K,J)=B(K,J)*ANFB	UFPO7330
GO TO 13	UFPO7340
12 A(K,J)=A(K,J)*BNFB	UFPO7350
B(K,J)=0.0	UFPO7360
13 AMP(K,J)=SQRT(A(K,J)**2+B(K,J)**2)	UFPO7370
PH(K,J)=ATAN2(A(K,J),B(K,J))*57.296	UFPO7380
IF(PH(K,J).LT.0.0) PH(K,J)=PH(K,J)+360.0	UFPO7390
PH(K,J)=PH(K,J)/NBLADE	UFPO7400
PH(1,J)=0.0	UFPO7410
AMP(1,J)=A(1,J)	UFPO7420
11 CONTINUE	UFPO7430
RETURN	UFPO7440
END	UFPO7450


```

SUBROUTINE PIEDAT(ISTOP)
*****
C  * SUBROUTINE PIEOAT GENERATES DATA FROM INPUT TO
C  * AXIAL VELOCITY OUTPUT TO BE USED AS INPUT FOR
C  * UNSTEADY PIEWAKE. IT USES BLADE GRID GEOMETRY
C  * TO CALCULATE NSR FIELD POINT AXIAL TIME-AVERAGED
C  * VELOCITIES AT FIVE RADII ALONG THE LEADING EDGE.
C  * IT THEN CALCULATES NSR FIELD POINT AXIAL TIME-
C  * AVERAGED VELOCITIES AT 1.1 TIMES PROPELLER RADIUS
C  *
C  * N      => CHORDWISE GRID COORDINATE
C  * M      => SPANWISE GRID COORDINATE
C  * K, KK  => ANGLE INCREMENTS + 1
C  * ISTOP=1 => STOP FPUV UPON RETURN TO MAIN PROG
C  * NSR    => NUMBER TIME STEPS IN REVOLUTION
C  * NR     => RADIUS NUMBER
C  * XFP,YBAR,ZBAR,RFP,TZFP
C  *      => X,Y,Z,RADIAL,THETA FIELD PT COORD
C  * THETA(1-5) => ANGLES 5 TIME STEPS BEFORE ZERO
C  *          BLADE ANGLE
C  * THETA(6 - NSR+5) => ANGLES IN ONE BLADE REV.
C  * THETA(NSR+6 - NSR+20) => ANGLES 15 TIME STEPS
C  *          AFTER ZERO BLADE ANGLE.
C  * VX(K)   => AXIAL VELOCITY AT ANGLE THETA(K)
C  * UA(KK,M) => SMOOTH AXIAL VELOCITY AT RADIUS M,
C  *          ANGLE KK
C  *
*****
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS
COMMON / GEDCDM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),
& XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),
& ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH
C *** COMMON TO CONTAIN SINGULARITY STRENGTHS
COMMON / SINCDM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV
& ,GAM(9,60),GMEAN(9),DKFW(20),HBWX,HBWY,HBWZ
C *** COMMON TO CONTAIN FIELD POINT DATA, NBLADE, IDENT, NSR, AND VSR
COMMON / FPCDM / NBLADE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
& DBLADE,DTFP,IDENT(18),TFP(60),VSR,IPL0T,DIAM
C *** COMMON TO CONTAIN VELOCITY DATA
COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP
C *** COMMON TO CONTAIN HARMONICS COEFFICIENTS

```

```

UFPO7460
UFPO7470
UFPO7480
UFPO7490
UFPO7500
UFPO7510
UFPO7520
UFPO7530
UFPO7540
UFPO7550
UFPO7560
UFPO7570
UFPO7580
UFPO7590
UFPO7600
UFPO7610
UFPO7620
UFPO7630
UFPO7640
UFPO7650
UFPO7660
UFPO7670
UFPO7680
UFPO7690
UFPO7700
UFPO7710
UFPO7720
UFPO7730
UFPO7740
UFPO7750
UFPO7760
UFPO7770
UFPO7780
UFPO7790
UFPO7800
UFPO7810
UFPO7820
UFPO7830
UFPO7840
UFPO7850
UFPO7860
UFPO7870
UFPO7880
UFPO7890
UFPO7900
UFPO7910
UFPO7920
UFPO7930
UFPO7940
UFPO7950
UFPO7960
UFPO7970
UFPO7980

```


C	COMMON / HRCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH	UFPO7990
C		UFPO8000
C		UFPO8010
C	DIMENSION THETA(80),VX(80),UA(60,9),RWK(9)	UFPO8020
	N=1	UFPO8030
	NR=1	UFPO8040
C		UFPO8050
C	*** CALCULATE VELOCITIES AT FIVE BLADE GRID RADII	UFPO8060
C		UFPO8070
C	102 00 300 M=3,9,2	UFPO8080
	NR=NR+1	UFPO8090
C		UFPO8100
C	*** CALCULATE VELOCITIES AT NSR ANGLES FOR EACH RADIUS	UFPO8110
C		UFPO8120
C	00 200 KK=1,NSR	UFPO8130
C		UFPO8140
C	*** FIELD POINT CENTERED ON BLADE GRID	UFPO8150
C		UFPO8160
	CALL FPGRID(X,Y,Z,N,M,KK,12)	UFPO8170
	IF (M.EQ.3) X1=XFP	UFPO8180
	KK15=KK+15	UFPO8190
	THETA(KK15)=TZFP	UFPO8200
	RWK(NR)=RFP	UFPO8210
	DTFP=360./FLOAT(NSR)	UFPO8220
C		UFPO8230
C	*** CALCULATE AXIAL VELOCITY FOR FIELD POINT AT RADIUS RWK(NR) AND	UFPO8240
C	*** ANGLE THETA(KK15). STORE THIS VELOCITY IN VX.	UFPO8250
C		UFPO8260
C	200 CALL AMPCAL(VX(16),KK)	UFPO8270
	NSR15=NSR+15	UFPO8280
C		UFPO8290
C	*** STORE SMOOTHED VELOCITIES IN NR-TH COLUMN OF UA	UFPO8300
C		UFPO8310
C	300 CALL SMOOTH(NSR,NR,VX,THETA,UA)	UFPO8320
C		UFPO8330
C	*** CALCULATE NSR VELOCITIES AT PROPELLER HUB	UFPO8340
C		UFPO8350
	NRHUB=1	UFPO8360
	XFP=1.5*X1	UFPO8370
	RWK(NRHUB)=RH	UFPO8380
C		UFPO8390
C	*** STEP THROUGH NSR ANGLES AT HUB	UFPO8400
C		UFPO8410
	DO 100 KK=1,NSR	UFPO8420
	KK15=KK+15	UFPO8430
	THETA(KK15)=360.*FLOAT(KK-1)/FLOAT(NSR)	UFPO8440
	TZFP=THETA(KK15)	UFPO8450
	100 CALL AMPCAL(VX(16),KK)	UFPO8460
	NSR15=NSR+15	UFPO8470
C		UFPO8480
C	*** NOW STORE SMOOTHED HUB VELOCITIES IN FIRST COLUMN OF UA	UFPO8490
C		UFPO8500
C		UFPO8510

C	CALL SMOOTH(NSR,NRHUB,VX,THETA,UA)	UFPO8520
C	*** NOW CALCULATE VELOCITIES AT A NEXT RADII OUTSIDE THE PROPELLER	UFPO8530
C	*** DISK. USE XFP OF PROPELLER GRID CASE OF N=1, M=9.	UFPO8540
C	&&& DIAGNOSTICS &&&&&&&&&&&&&&&&	UFPO8550
C	GOTO 101	UFPO8560
C	DO 500 KKK=1,7,3	UFPO8570
C	RFP=1.O+FLOAT(KKK)/10.	UFPO8580
C	NR=NR+1	UFPO8590
C	RWK(NR)=RFP	UFPO8600
C	*** CALCULATE NSR VELOCITIES AT 1.1,1.4, AND 1.7 PROP RADII	UFPO8610
C	DO 400 KK=1,NSR	UFPO8620
C	KK15=KK+15	UFPO8630
C	THETA(KK15)=360.*FLOAT(KK-1)/FLOAT(NSR)	UFPO8640
C	TZFP=THETA(KK15)	UFPO8650
C	400 CALL AMPCAL(VX(16),KK)	UFPO8660
C	*** NOW STORE SMOOTHED VELOCITIES IN NR-TH COLUMN OF UA	UFPO8680
C	500 CALL SMOOTH(NSR,NR,VX,THETA,UA)	UFPO8690
C	*** WRITE(OUTPUT IN FORMAT FOR UNSTEADY PIEWAKE	UFPO8700
C	901 FORMAT(8F10.5)	UFPO8710
C	902 FORMAT(16I5)	UFPO8720
C	101 WRITE(8,902) NR	UFPO8730
C	WRITE(8,901) (RWK(M),M=1,NR)	UFPO8740
C	WRITE(8,901) ((UA(KK,M),KK=1,NSR),M=1,NR)	UFPO8750
C	*** STOP FPUV UPON RETURN TO MAIN PROGRAM	UFPO8760
C	ISTOP=1	UFPO8770
C	RETURN	UFPO8780
C	END	UFPO8790
		UFPO8800
		UFPO8810
		UFPO8820
		UFPO8830
		UFPO8840
		UFPO8850
		UFPO8860
		UFPO8870
		UFPO8880
		UFPO8890
		UFPO8900
		UFPO8910
		UFPO8920
		UFPO8930

C	SUBROUTINE AMPCAL(VX, KK)	UFPO8940
C	*****	UFPO8950
C	*	UFPO8960
C	* SUBROUTINE AMPCAL WILL TAKE FIELDO POINT GEOMETRY *	UFPO8970
C	* CALCULATED IN PIEOAT AND CALCULATE THE INDUCED *	UFPO8980
C	* TIME-AVERAGED AXIAL VELOCITY. IT WILL STORE THAT *	UFPO8990
C	* VELOCITY IN VX(KK).	UFPO9000
C	*	UFPO9010
C	* NSR => NUMBER OF ANGLES PER REVOLUTION *	UFPO9020
C	* K => ANGLE INCREMENTS + 1 *	UFPO9030
C	* VX(KK) => WHERE TIME-AVERAGED AXIAL VEL STORED *	UFPO9040
C	* AMP(1,1) => ZEROETH-HARMONIC AXIAL VELOCITY *	UFPO9050
C	* COMPUTED BY FOUICAL *	UFPO9060
C	*	UFPO9070
C	*****	UFPO9080
C		UFPO9090
C		UFPO9100
C		UFPO9110
C	*** COMMON TO CONTAIN GEOMETRIC PARAMETERS	UFPO9120
C		UFPO9130
C	COMMON / GEODCOM / MM, MN, NN, NW, NTOPL, X(12, 10), Y(12, 10), Z(12, 10),	UFPO9140
C	& XTIP(10, 11), YTIP(10, 11), ZTIP(10, 11), XH(101, 7), YH(101, 7),	UFPO9150
C	& ZH(101, 7), XW(21, 10), YW(21, 10), ZW(21, 10), RH	UFPO9160
C		UFPO9170
C	*** COMMON TO CONTAIN SINGULARITY STRENGTHS	UFPO9180
C		UFPO9190
C	COMMON / SINCOM / GT(10, 10, 60), GB(10, 9, 60), GTW(10), SB(10, 9), GTV	UFPO9200
C	& , GAM(9, 60), GMEAN(9), OKFW(20), HBWX, HBWY, HBWZ	UFPO9210
C		UFPO9220
C	*** COMMON TO CONTAIN FIELDO POINT DATA, NBLAOE, IDENT, NSR, AND VSR	UFPO9230
C		UFPO9240
C	COMMON / FPCOM / NBLAOE, NSR, XFP, RFP, TZFP, YBAR, ZBAR,	UFPO9250
C	& OBLAOE, OTFP, IDENT(18), TFP(60), VSR, IPLOT, OIAM	UFPO9260
C		UFPO9270
C	*** COMMON TO CONTAIN VELOCITY DATA	UFPO9280
C		UFPO9290
C	COMMON / VELCOM / VIX(60), VIT(60), VIR(60), U(60, 3), NSAMP	UFPO9300
C		UFPO9310
C	*** COMMON TO CONTAIN HARMONICS COEFFICIENTS	UFPO9320
C		UFPO9330
C	COMMON / HARCOT / A(15, 3), B(15, 3), AMP(15, 3), PH(15, 3), NH	UFPO9340
C		UFPO9350
C	DIMENSION VX(1)	UFPO9360
C		UFPO9370
C	*** RUN THROUGH AND FPSTEP FOR EACH TIME STEP	UFPO9380
C		UFPO9390
C	DO 100 K=1, NSR	UFPO9400
C		UFPO9410
C	*** FPSTEP CALCULATES VELOCITY INDUCED AT A FIELDO POINT BY ONE	UFPO9420
C	*** BLAOE AT TIME STEP K	UFPO9430
C		UFPO9440
C	100 CALL FPSTEP(K)	UFPO9450
C		UFPO9460

C *** SUMVEL SUMS FIELD POINT VELOCITIES INDUCED BY EACH BLADE	UFPO9470
C	UFPO9480
CALL SUMVEL	UFPO9490
C	UFPO9500
C *** FOUCAL GENERATES HARMONIC COEFFICIENTS FOR INDUCED FIELD POINT	UFPO9510
C *** VELOCITY AS PROPELLER ROTATES.	UFPO9520
C	UFPO9530
CALL FOUCAL	UFPO9540
C	UFPO9550
C *** STORE TIME-AVERAGED (ZERO-HARMONIC) VELOCITY IN VX	UFPO9560
C	UFPO9570
VX(KK)=AMP(1,1)	UFPO9580
RETURN	UFPO9590
END	UFPO9600

C	SUBROUTINE SMOOTH(NSR,NR,VX,THETA,UA)	UFP09610
C	*****	UFP09620
C	*	UFP09630
C	* SUBROUTINE SMOOTH READS INOUCED VELOCITIES AT ONE*	UFP09640
C	* RAOIUS FROM PIEDAT. IT CALCULATES BY	UFP09650
C	* INTERPOLATION A SERIES OF NSR EVENLY SPACED	UFP09660
C	* AXIAL INDUCED VELOCITIES SUITABLE FOR INPUT INTO	UFP09670
C	* UNSTEADY PIEWAKE.	UFP09680
C	*	UFP09690
C	* NSR => NUMBER OF ANGLES PER REVOLUTION	UFP09700
C	* NR => RAOIUS NUMBER 1=HUB, ETC.	UFP09710
C	* VX(1 - 5) => RAW VELOCITIES 5 STEPS BEFORE ZERO	UFP09720
C	* BLADE ANGLE.	UFP09730
C	* VX(6 - NSR+5) => RAW VELOCITIES IN ONE REVOLUTION*	UFP09740
C	* VX(NSR+6 - NSR+20) => RAW VELOCITIES 15 TIME	UFP09750
C	* TIME STEPS AFTER ONE REVOLUTION	UFP09760
C	* THETA(K) => ANGLE OF VELOCITY VX(K)	UFP09770
C	* UA(K,M) => SMOOTH VEL AT RAOIUS M, ANGLE K	UFP09780
C	* TH(K) => EVENLY SPACED ANGLES OF UA(K,M)	UFP09790
C	* NIN => NUMBER OF ANGLES FED TO UGLYOK	UFP09800
C	* KPOS => K OF FIRST POS. ANGLE IN THETA(K)	UFP09810
C	* KSTART,KSTOP => FIRST AND LAST INOEX OF THETA(K)	UFP09820
C	* AND VX(K) SENT TO UGLYOK	UFP09830
C	* AVX => SPLINE CURVE COEFFICIENTS OF VX	UFP09840
C	*	UFP09850
C	*****	UFP09860
C		UFP09870
C		UFP09880
C	DIMENSION VX(80),UA(60,9),THETA(80),TH(60),AVX(250)	UFP09890
C	*** COPY LAST 15 STEPS IN REV INTO FIRST 15 ENTRIES IN VX AND THETA	UFP09900
C		UFP09910
C	00 100 K=1,15	UFP09920
C	KNSR=K+NSR	UFP09930
C	THETA(K)=THETA(KNSR)-360.	UFP09940
C	100 VX(K)=VX(KNSR)	UFP09950
C		UFP09960
C		UFP09970
C	*** COPY FIRST 5 TIME STEPS IN REV INTO 5 ENTRIES AFTER REVOLUTION	UFP09980
C		UFP09990
C	00 200 K=16,20	UFP10000
C	KNSR=K+NSR	UFP10010
C	THETA(KNSR)=THETA(K) + 360.	UFP10020
C	200 VX(KNSR)=VX(K)	UFP10030
C		UFP10040
C	*** SET UP EVENLY SPACED ANGLES	UFP10050
C		UFP10060
C	DO 300 K=1,NSR	UFP10070
C	300 TH(K)=360.*FLOAT(K-1)/FLOAT(NSR)	UFP10080
C	901 FORMAT(8F10.5)	UFP10090
C	902 FORMAT(16I5)	UFP10100
C		UFP10110
C	*** FIND FIRST POSITIVE ANGLE, SAVE INDEX	UFP10120
C		UFP10130

DO 400 K=1,NSR	UFP10140
KPOS=K	UFP10150
IF (THETA(K).GT.O.) GOTO 500	UFP10160
400 CONTINUE	UFP10170
500 KSTART=KPOS-1	UFP10180
NSR2=NSR/2	UFP10190
KSTOP=KPOS+NSR2+1	UFP10200
NIN=KSTOP-KSTART+1	UFP10210
C	UFP10220
C *** FINE SPLINE COEFFICIENTS AND SMOOTH VEL FOR FIRST 180 DEGREES	UFP10230
C	UFP10240
CALL UGLYDK(NIN,1,1,THETA(KSTART),VX(KSTART),O.,O.,AVX)	UFP10250
CALL EVALDK(NIN,NSR2,THETA(KSTART),TH(1),UA(1,NR),AVX)	UFP10260
C	UFP10270
C *** FIND SPLINE COEFFICIENTS AND SMOOTH VEL FOR LAST 180 DEGREES	UFP10280
C	UFP10290
KSTART=KSTOP-2	UFP10300
KSTOP=KSTART+NIN-1	UFP10310
NSR2P=NSR2+1	UFP10320
CALL UGLYDK(NIN,1,1,THETA(KSTART),VX(KSTART),O.,O.,AVX)	UFP10330
CALL EVALDK(NIN,NSR2,THETA(KSTART),TH(NSR2P),UA(NSR2P,NR),AVX)	UFP10340
RETURN	UFP10350
END	UFP10360

** APPENDIX XI **

PIEWAKE Program Listing


```

C      THIS PROGRAM COMPUTES EFFECTIVE WAKES BY ONV METHOD
C
C      *-----*
C      *
C      *   THE ORIGINAL PIEWAKE WAS WRITTEN BY R. VAN HOUTEN
C      *   IN ABOUT 1981. IT IS BASED ON THE OET NORSKE
C      *   VERITAS METHOD OF COMPUTING EFFECTIVE WAKE. THIS
C      *   PROGRAM HAS BEEN MODIFIED BY R. JAMISON IN APRIL,
C      *   1982, TO ACCOMMODATE UNSTEADY PROPELLER-INDUCED
C      *   AXIAL VELOCITIES FROM UFPV. BASICALLY, BOTH
C      *   PROGRAMS DIVIDE THE WAKE INTO A NUMBER OF PIE-
C      *   SHAPED SEGMENTS. EACH SEGMENT IS ASSUMED TO
C      *   CONTRACT IN AN AXISYMMETRIC MANNER AS GIVEN BY
C      *   T. HUANG.
C      *
C      *   NP      => NUMBER OF PIE-SHAPED SEGMENTS
C      *   NX      => NUMBER OF NOMINAL WAKE RAOII
C      *   RX(M)   => NOMINAL WAKE RADIUS M (UP TO 11)
C      *   VO(K,M) => NOMINAL AXIAL VELOCITY, SEG K, RAO M
C      *   VT(K,M) => NOM TANG VEL, SEG K, RAO M
C      *   VR(K,M) => NOM RAO VEL, SEG K, RAO M
C      *   VTE,VRE => VT & VR INTERPOLATED AT EFF WAKE RAO
C      *   UASEG(M) => UA IN A PIE-SEGMENT. ALSO USED TO
C      *               STORE A PIE-SEGMENT OF VT
C      *   UX(M)   => VO IN A PIE-SEGMENT. ALSO USED TO
C      *               STORE A PIE-SEGMENT OF VR
C      *   AUX,UASEG=> SPLINE COEFF. FOR VR,VT, RESP.
C      *   RE(M)   => RAOII OF EFFECTIVE WAKE VELOCITIES
C      *   THETA(K) => ANGLE IN DEGREES OF PIE SEG K
C      *   NA      => NUMBER OF PROP-INO, AXIAL VEL RAOII
C      *   RA(M)   => PROP-INO AXIAL VEL RADIUS M
C      *   UA(K,M) => PROP-INO AXIAL VEL, SEG K, RAO M
C      *   VE(K,M) => EFFECTIVE WAKE VEL, SEG K, RAO M
C      *   VL      => AVERAGE VELOCITY
C      *   VLAVN   => VOLUMETRIC AVERAGE NOMINAL VELOCITY
C      *   VLAVE    => VOLUMETRIC AVERAGE EFFECTIVE VELOCITY
C      *   BLOCK   => EFFECTIVE BLOCKAGE
C      *-----*
C      DIMENSION VO(60,11),RX(11),UX(60),RA(11),UA(60,11),UASEG(60)
C      DIMENSION RE(11), VR(60,11),VT(60,11),UASEG(44),AUX(44),VTE(11)
C      DIMENSION THETA(60),VE(60,11),IDENT(18),UERX(60),VRE(11)
C      COMMON RX,UX,RA,UASEG,UERX,RE
C      110 FORMAT(8F10.5)
C      103 FORMAT('/' AVERAGE VELOCITY =',F5.3)
C      113 FORMAT('/' VOLUMETRIC AVERAGE NOMINAL VELOCITY =',F5.3)
C      114 FORMAT('/' VOLUMETRIC AVERAGE EFFECTIVE VELOCITY =',F5.3)
C      115 FORMAT('/' EFFECTIVE BLOCKAGE =',F7.3,' PERCENT')
C      111 FORMAT(18A4)
C      210 FORMAT(16I5)
C      200 FORMAT(8F10.5)
C
C      *** READ IN NOMINAL WAKE DATA. SAVE RAOIAL AND TANGENTIAL
C
PIE00010
PIE00020
PIE00030
PIE00040
PIE00050
PIE00060
PIE00070
PIE00080
PIE00090
PIE00100
PIE00110
PIE00120
PIE00130
PIE00140
PIE00150
PIE00160
PIE00170
PIE00180
PIE00190
PIE00200
PIE00210
PIE00220
PIE00230
PIE00240
PIE00250
PIE00260
PIE00270
PIE00280
PIE00290
PIE00300
PIE00310
PIE00320
PIE00330
PIE00340
PIE00350
PIE00360
PIE00370
PIE00380
PIE00390
PIE00400
PIE00410
PIE00420
PIE00430
PIE00440
PIE00450
PIE00460
PIE00470
PIE00480
PIE00490
PIE00500
PIE00510
PIE00520
PIE00530

```


C	VELDCITIES. THEY ARE NOT NEEDED FOR EFFECTIVE WAKE	PIE00540
C	CALCULATIONS BUT WILL BE ECHOED FOR WKPRDC.	PIE00550
C		PIE00560
	READ(10,111) (IOENT(I),I=1,18)	PIE00570
	READ(10,210) NX	PIE00580
	READ(10,210) NP	PIE00590
	READ(10,200)(RX(M),M=1,NX)	PIE00600
	DO 20 M=1,NX	PIE00610
	READ(10,200)((THETA(K),VD(K,M),VT(K,M),VR(K,M)),K=1,NP)	PIE00620
	20 CONTINUE	PIE00630
C		PIE00640
C	*** READ IN PROPELLER-INDUCED VELOCITIES FROM EITHER FPUV	PIE00650
C	OR A SEPARATE DATA FILE	PIE00660
C		PIE00670
	900 FDMAT('/' DO YOU WISH TO USE STEADY OR UNSTEADY VELDCITY PROFILE?'	PIE00680
	& '/' ENTER 1 FOR STEADY, 2 FOR UNSTEADY')	PIE00690
	8 WRITE(6,900)	PIE00700
	READ(5,*) ISTEAD	PIE00710
	IF (ISTEAD.GT.2.DR.ISTEAD.LT.1) GOTO 8	PIE00720
	READ(8,210) NA	PIE00730
	READ(8,200) (RA(I),I=1,NA)	PIE00740
	RH=RA(1)	PIE00750
	RPROP=1.	PIE00760
	IF(ISTEAD.EQ.1) GOTO 22	PIE00770
	READ(8,200) ((UA(KK,M),KK=1,NP),M=1,NA)	PIE00780
	GOTO 92	PIE00790
	22 READ(8,200) (UA(1,M),M=1,NA)	PIE00800
	DO 21 KK=2,NP	PIE00810
	DO 21 M=1,NA	PIE00820
	21 UA(KK,M)=UA(1,M)	PIE00830
C		PIE00840
C	*** EXTRAPOLATE NDMINAL WAKE DATA TO HUB	PIE00850
C		PIE00860
	92 WRITE(6,100)	PIE00870
	100 FDMAT('/' DO YOU WISH TO EXTRAPOLATE VELDCITY DATA TO THE HUB?'/	PIE00880
	1 ' ENTER 2 FOR NO, 1 FOR LINEAR, 0 FOR CONSTANT')	PIE00890
	READ(5,*) IXTAP	PIE00900
	NX1=NX	PIE00910
	IF(IXTAP.GT.1)GO TO 28	PIE00920
C		PIE00930
C	*** LINEAR EXTRAPOLATION OF NDMINAL WAKE DATA TO HUB.	PIE00940
C		PIE00950
	NX=NX+1	PIE00960
	DO 25 K=1,NP	PIE00970
	DO 23 M=1,NX1	PIE00980
	MR=NX-M	PIE00990
	MRP1=MR+1	PIE01000
	VT(K,MRP1)=VT(K,MR)	PIE01010
	VR(K,MRP1)=VR(K,MR)	PIE01020
	23 VD(K,MRP1)=VD(K,MR)	PIE01030
	25 VO(K,1)=VD(K,2)+IXTAP*(VO(K,2)-VO(K,3))/(RX(1)-RX(2))*(RH-RX(1))	PIE01040
	DO 26 M=1,NX1	PIE01050
	MR=NX-M	PIE01060

MRP1=MR+1	PIEO1070
26 RX(MRP1)=RX(MR)	PIEO1080
RX(1)=RH	PIEO1090
28 VL=0.	PIEO1100
VLAVN=0.	PIEO1110
VLAVE=0.	PIEO1120
C	PIEO1130
C *** CYCLE THROUGH EACH PIE SEGMENT	PIEO1140
C	PIEO1150
DO 50 K=1,NP	PIEO1160
DO 30 M=1,NX	PIEO1170
UASEG(M)=UA(K,M)	PIEO1180
30 UX(M)=VO(K,M)	PIEO1190
C	PIEO1200
C *** WKMOD PERFORMS THE HUANG MODIFICATION ON ONE PIE-SEGMENT	PIEO1210
C	PIEO1220
CALL WKMOD(NX,NX,NA,1,RPROP,VOL,VOLAVN,VOLAVE)	PIEO1230
C	PIEO1240
C *** INTERPOLATE VT AND VR AT EFFECTIVE WAKE RADII	PIEO1250
C	PIEO1260
DO 60 M=1,NX	PIEO1270
UASEG(M)=VT(K,M)	PIEO1280
60 UX(M)=VR(K,M)	PIEO1290
CALL UGLYDK(NX,1,1,RX,UASEG,O.,O.,AUASEG)	PIEO1300
CALL UGLYDK(NX,1,1,RX,UX,O.,O.,AUX)	PIEO1310
CALL EVALDK(NX,NX,RX,RE,VTE,AUASEG)	PIEO1320
CALL EVALDK(NX,NX,RX,RE,VRE,AUX)	PIEO1330
DO 70 M=1,NX	PIEO1340
VT(K,M)=VTE(M)	PIEO1350
70 VR(K,M)=VRE(M)	PIEO1360
C	PIEO1370
C *** ACCUMULATE VOLUMETRIC AVERAGE DATA	PIEO1380
C	PIEO1390
VL=VL+VOL	PIEO1400
VLAVN=VLAVN+VOLAVN	PIEO1410
VLAVE=VLAVE+VOLAVE	PIEO1420
DO 40 M=1,NX	PIEO1430
40 VE(K,M)=UERX(M)	PIEO1440
50 CONTINUE	PIEO1450
C	PIEO1460
C *** COMPUTE VOLUMETRIC AVERAGES	PIEO1470
C	PIEO1480
VL=VL/NP	PIEO1490
VLAVN=VLAVN/NP	PIEO1500
VLAVE=VLAVE/NP	PIEO1510
BLOCK=(1.-VLAVE/VLAVN)*100.	PIEO1520
C	PIEO1530
C *** WRITE EFFECTIVE WAKE DATA IN FILE 11 TO BE PROCESSED BY WKPROC	PIEO1540
C	PIEO1550
WRITE(11,111) (IDENT(I),I=1,18)	PIEO1560
WRITE(11,210) NX	PIEO1570
WRITE(11,210) (NP,M=1,NX)	PIEO1580
WRITE(11,200)(RE(M),M=1,NX)	PIEO1590


```

      DO 120 M=1,NX
      WRITE(11,200)((THETA(K),VE(K,M),VT(K,M),VR(K,M)),K=1,NP)
120  CONTINUE
C
C *** WRITE AVERAGES TO TERMINAL
C
      WRITE(6,103)VL
      WRITE(6,113)VLAVN
      WRITE(6,114)VLAVE
      WRITE(6,115)BLOCK
      STOP
      END

```

```

PIEO1600
PIEO1610
PIEO1620
PIEO1630
PIEO1640
PIEO1650
PIEO1660
PIEO1670
PIEO1680
PIEO1690
PIEO1700
PIEO1710

```


C	SUBROUTINE WKMOD(N,NX,NA,IOW,RPROP,VOL,VOLAVN,VOLAVE)	PIEO1720
C	*****	PIEO1730
C	*	PIEO1740
C	* SUBROUTINE WKMOD CALCULATES THE THOMAS HUANG	PIEO1750
C	* CONTRACTION OF THE NOMINAL AND INDUCED WAKES	PIEO1760
C	* PASSED TO IT THROUGH COMMON. IT ALSO COMPUTES	PIEO1770
C	* VOLUMETRIC AVERAGES.	PIEO1780
C	*	PIEO1790
C	*	PIEO1800
C	* NX => NUMBER OF NOMINAL RADII	PIEO1810
C	* RH => RADIUS OF HUB	PIEO1820
C	* RT => RADIUS OF OUTERMOST NOMINAL VELOCITY	PIEO1830
C	* UX,UXR => NOMINAL WAKE VELOCITIES	PIEO1840
C	* UE,UEXR => EFFECTIVE WAKE VELOCITIES	PIEO1850
C	* UP => APPARENT WAKE = EFF. WAKE + IND. WAKE	PIEO1860
C	* UA,UAR => INDUCED VELOCITIES	PIEO1870
C	* IOW = 1 => NO TUNNEL CORRECTIONS	PIEO1880
C	* ITER => OPEN WATER ITERATION INOEX	PIEO1890
C	* ITERAT => HUANG CORRECTION FACTOR INOEX	PIEO1900
C	* B,C,D,F => HUANG CORRECTION FACTORS. SEE PAPER	PIEO1910
C	* AUA,AUX,AUE=> SPLINE COEFF. FOR INDUCED, NOMINAL,	PIEO1920
C	* AND EFFECTIVE VELOCITY FIELDS. RESP	PIEO1930
C	* RPROP => RADIUS OF PROPELLER	PIEO1940
C	* RE(M) => SPECIFIED EFFECTIVE WAKE RADII	PIEO1950
C	* VDL => AVERAGE VELOCITY	PIEO1960
C	* VLAVN => VOLUMETRIC AVERAGE NOMINAL VELOCITY	PIEO1970
C	* VLAVE => VOLUMETRIC AVERAGE EFFECTIVE VELOCITY	PIEO1980
C	*	PIEO1990
C	*****	PIEO2000
	DIMENSION RX(11),R(11),RA(11),UX(60),UXR(60),UA(60),UP(60),UE(60)	PIEO2010
	DIMENSION RE(11),UAR(60),RP(11)	PIEO2020
	DIMENSION AUX(240),AUA(240)	PIEO2030
	DIMENSION AUE(240),UERX(60)	PIEO2040
	COMMON RX,UX,RA,UA,UERX,RE	PIEO2050
C		PIEO2060
C	*** USE R(I) FOR NOMINAL RADII	PIEO2070
C		PIEO2080
	N=NX	PIEO2090
	N1=N-1	PIEO2100
	RH=RX(1)	PIEO2110
	RT=RX(NX)	PIEO2120
C		PIEO2130
C	*** INTERPOLATE TO FIND NOMINAL VELOCITIES AT NOMINAL RADII	PIEO2140
C		PIEO2150
	CALL UGLYDK(NX,1,1,RX,UX,O.,O.,AUX)	PIEO2160
	DO 10 I=1,N	PIEO2170
C	10 R(I)=(RT-RH)/(N-1)*(I-1)+RH	PIEO2180
	10 R(I)=RX(I)	PIEO2190
	CALL EVALOK(NX,N,RX,R,UXR,AUX)	PIEO2200
C		PIEO2210
C	*** INTERPOLATE TO FIND INDUCED VELOCITIES AT NOMINAL RADII	PIEO2220
C		PIEO2230
	CALL UGLYDK(NA,1,1,RA,UA,O.,O.,AUA)	PIEO2240

CALL EVALOK(NA,N,RA,R,UAR,AUA)	PIE02250
901 FORMAT(8F10.5)	PIE02260
C	PIE02270
C *** OUTSIDE PROPELLER DISK, NOMINAL WAKE = EFFECTIVE WAKE	PIE02280
C	PIE02290
IUPDAT=0	PIE02300
ITER=1	PIE02310
ITERAT=1	PIE02320
UE(N)=UXR(N)	PIE02330
20 UP(N)=UE(N)+UAR(N)	PIE02340
C	PIE02350
C *** ESTIMATE EFF. WAKE MARCHING FROM OUTSIDE TO HUB	PIE02360
C	PIE02370
DO 30 I=1,N1	PIE02380
K=N-I	PIE02390
UE(K)=SQRT((UE(K+1)+(UAR(K+1)+UAR(K))/2.)*2+UXR(K)*2-UXR(K+1)*2-	PIE02400
1) -(UAR(K+1)+UAR(K))/2.	PIE02410
UP(K)=UE(K)+UAR(K)	PIE02420
30 CONTINUE	PIE02430
C	PIE02440
C *** COMPUTE HUANG'S CORRECTION FACTORS	PIE02450
C	PIE02460
RP(1)=RH	PIE02470
VOL=0.	PIE02480
DO 40 K=1,N1	PIE02490
B=2*UP(K+1)+UP(K)	PIE02500
C=-RP(K)*(UP(K+1)-UP(K))	PIE02510
F=(R(K+1)*2-R(K)*2)*(2*UXR(K+1)+UXR(K))-R(K)*(UXR(K+1)-UXR(K))*	PIE02520
1R(K+1)-R(K))	PIE02530
VOL=VOL+F	PIE02540
D=-RP(K)*2*(UP(K+1)+2*UP(K))-F	PIE02550
C	PIE02560
C *** FIND RADIUS FOR (K+1)ST APPARENT WAKE	PIE02570
C	PIE02580
IF ((C=C-4*B*D).LT.0.) GOTO 80	PIE02590
40 RP(K+1)=(-C+SQRT(C*2-4*B*D))/2./B	PIE02600
IF (ABS(RP(N)-RT).LT.(.0001*RT))GO TO 60	PIE02610
IUPDAT=0	PIE02620
C	PIE02630
C *** IOW=0 => TUNNEL CORRECTIONS	PIE02640
C	PIE02650
=1 => OPEN WATER (I.E., INFINITE FLUID)	PIE02660
C	PIE02670
IF(IOW.EQ.1) GO TO 60	PIE02680
IF (ITER.NE.1)GO TO 50	PIE02690
ITER=2	PIE02700
UENOLD=UE(N)	PIE02710
RPNOLD=RP(N)	PIE02720
UE(N)=UE(N)*.98	PIE02730
GO TO 20	PIE02740
50 CONTINUE	PIE02750
ITER=ITER+1	PIE02760
IF(ITER.GT.10)GO TO 80	PIE02770
UEN=UE(N)	

UE(N)=UE(N)+(RT-RP(N))/(RP(N)-RPNDLD)*(UE(N)-UENDLD)	PIE02780
RPNDLD=RP(N)	PIE02790
UENDLD=UEN	PIE02800
GD TO 20	PIE02810
C	PIE02820
C *** ITERATE ON HUANG CORRECTION FACTORS, 3 TIMES FOR OPEN WATER	PIE02830
C	PIE02840
60 IF(IUPDAT.EQ.1)GD TO 70	PIE02850
IF(IDW.EQ.1.AND.ITERAT.EQ.5)GD TO 70	PIE02860
ITER=1	PIE02870
ITERAT=ITERAT+1	PIE02880
IF(ITERAT.GT.10)GD TO 80	PIE02890
IUPDAT=1	PIE02900
C	PIE02910
C *** FIND INDUCED VELOCITIES AT APPARENT WAKE RADII	PIE02920
C	PIE02930
CALL EVALDK(NA,N,RA,RP,UAR,AUA)	PIE02940
GD TO 20	PIE02950
70 CONTINUE	PIE02960
C	PIE02970
C *** CALCULATE EFFECTIVE WAKE RADII	PIE02980
C	PIE02990
NXM1=NX-1	PIE03000
DD 100 M=1,NX	PIE03010
100 RE(M)=RH+(RPRDP-RH)*FLDAT(M-1)/FLDAT(NXM1)	PIE03020
C	PIE03030
C *** IF DUTERMDST APPARENT RADIUS IS INSIDE PROPELLER DISK, THEN	PIE03040
C LINEARLY EXTRAPOLATE TO PROPELLER RADIUS	PIE03050
C	PIE03060
IF (RP(N).GE.RPRDP) GOTO 110	PIE03070
UE(N)=UE(N1) + (UE(N)-UE(N1))*(RPRDP-R(N1))/(RP(N)-RP(N1))	PIE03080
RP(N)=RPRDP	PIE03090
C	PIE03100
C *** INTERPOLATE EFFECTIVE WAKE VELOCITIES AT SPECIFIED RADII	PIE03110
C	PIE03120
110 CALL UGLYDK(N,1,1,RP,UE,O.,O.,AUE)	PIE03130
CALL EVALDK(N,NX,RP,RE,UERX,AUE)	PIE03140
C	PIE03150
C *** COMPUTE VOL = AVERAGE VELOCITY	PIE03160
C	PIE03170
VOL=VOL/3./((RT**2-RH**2)	PIE03180
C	PIE03190
C *** COMPUTE VOLUMETRIC AVERAGE NOMINAL VELOCITY	PIE03200
C	PIE03210
CALL INTEDK(NX,RX,RH,RPRDP,YDX,XYDX,XXYDX,AUX)	PIE03220
VOLAVN=XYDX*2./((RPROP**2-RH**2)	PIE03230
C	PIE03240
C *** COMPUTE VOLUMETRIC AVERAGE EFFECTIVE VELOCITY	PIE03250
C	PIE03260
CALL INTEDK(N,RP,RH,RPRDP,YDX,XYDX,XXYDX,AUE)	PIE03270
VOLAVE=XYDX*2./((RPROP**2-RH**2)	PIE03280
RETURN	PIE03290
C	PIE03300

C *** TUNNEL CORRECTION OR HUANG CORRECTION ITERATIONS FAIL TO CONVERGE P1E03310
C P1E03320
80 STOP P1E03330
END P1E03340

REFERENCES

- (1) Boswell, Robert J., "Design, Cavitation Performance, and Open-Water Performance of a Series of Research Skewed Propellers", Naval Research and Development Center Report No. 3339, 1971
- (2) Valentine, D.T., and Chase, A., "Highly Skewed Propeller Design for a Naval Auxiliary Oiler (AO-177)", Naval Research and Development Center Report No. SPD-544-12, 1976
- (3) Hendrican, A., and Remmers, K., "Powering and Cavitation Performance for a Naval Fleet Oiler, AO-177 Class (Model 5326 with Propeller 4677)", Naval Research and Development Center Report No. SPD-544-14, 1976
- (4) Burrill, L.C., and Emerson, A., "Propeller Cavitation: Further Tests on 16-inch Propellers in the Kings College Cavitation Tunnel", Northeast Coast Institute of Engineers and Shipbuilders, Vol. 79, pp. 295-320, 1962-63
- (5) Kerwin, J.E., and Lee, C.-S., "Prediction of Steady and Unsteady Marine Propeller Performance by Numerical Lifting-Surface Theory", SNAME Transactions, Vol. 86, pp. 218-253, 1978
- (6) Min, K.-S., "Numerical and Experimental Methods for

the Prediction of Field Point Velocities Around Propeller Blades", Department of Ocean Engineering of Massachusetts Institute of Technology Report No. 78-12, 1978

(7) Lee, C.-S., "Prediction of Steady and Unsteady Performance of Marine Propellers with or without Cavitation by Numerical Lifting-Surface Theory", Ph.D. Thesis, MIT, 1979

(8) Huang, T.T., et al., "Propeller/Stern/Boundary-Layer Interaction on Axisymmetric Bodies: Theory and Experiment", DTNSRDC Report No. 76-0113, 1976

(9) Huang, T.T., and Groves, N.C., "Effective Wake: Theory and Experiment", 13th Symposium on Naval Hydrodynamics, Tokyo, Japan, 1980

(10) Bjarne, E., "US Navy Oiler (AO-177 Class) - Model Tests in SSPA Cavitation Tunnel #2 - Design Propeller", Swedish Maritime Research Center, Goteborg, Sweden, DTNSRDC Report No. 2564-1, 1980

Thesis

J2725

c.1

Jamison

199476

Evaluation of a
highly skewed propeller
for a Naval auxiliary
(AO-177).

Thesis

J2725

c.1

Jamison

199476

Evaluation of a
highly skewed propeller
for a Naval auxiliary
(AO-177).

thesJ2725

Evaluation of a highly skewed propeller



3 2768 002 10014 1

DUDLEY KNOX LIBRARY